On Supporting K-anonymisation and L-diversity of Crime Databases with Genetic Algorithms in a Resource Constrained Environment

---

Submitted for Examination by

C.T. Verster

Student Number: VRSCOR001

Supervised by:

Dr. Anne Kayem

Department of Computer Science

January 2015

A dissertation submitted to the Department of Computer Science,

University of Cape Town,

in partial fulfilment of the requirements

for the degree of

Master of Science in Information Technology

Declaration

I know the meaning of plagiarism and declare that all the work in this thesis, save for that which is properly acknowledged, is my own.

Signature of Author .....

Cape Town
29 January 2015

Abstract

The social benefits derived from analysing crime data need to be weighed against issues relating to privacy loss. To facilitate such analysis of crime data Burke and Kayem [7] proposed a framework (MCRF) to enable mobile crime reporting in a developing country. Here crimes are reported via mobile phones and stored in a database owned by a law enforcement agency. The expertise required to perform analysis on the crime data is however unlikely to be available within the law enforcement agency. Burke and Kayem [7] proposed anonymising the data(using manual input parameters) at the law enforcement agency before sending it to a third party for analysis. Whilst analysis of the crime data requires expertise, adequate skill to appropriately anonymise the data is also required. What is lacking in the original MCRF is therefore an automated scheme for the law enforcement agency to adequately anonymise the data before sending it to the third party. This should, however, be done whilst maximising information utility of the anonymised data from the perspective of the third party.

In this thesis we introduce a crime severity scale to facilitate the automation of data anonymisation within the MCRF. We consider a modified loss metric to capture information loss incurred during the anonymisation process. This modified loss metric also gives third party users the flexibility to specify attributes of the anonymised data when requesting data from the law enforcement agency. We employ a genetic algorithm(GA) approach called "CrimeGenes"(CG) to optimise utility of the anonymised data based on our modified loss metric whilst adhering to notions of privacy defined by k-anonymity and l-diversity. Our CG implementation is modular and can therefore be easily integrated with the original MCRF. We also show how our CG approach is designed to be suitable for implementation in a developing country where particular resource constraints exist.

Acknowledgements

Thank you to my wife, Leandra, for looking after the children and helping me to make time to write this paper. My supervisor, Dr Anne Kayem, thank you for your guidance and patience.

Contents

List of Figures

Chapter 1

Introduction

This thesis explores avenues of automated data anonymisation for crime databases in a setting where limitations on human technical expertise, computational capacity and time constraints impede the use of standard solutions. These limitations are characteristic of a developing country. Our focus on crime data is well suited to the inherent trade-off that exists between data utility and privacy when publishing anonymised data. There is a social benefit when existing incidents of crime can be analysed to help reduce future criminal activity. However this needs to be weighed against the fact that such analysis causes a risk of privacy loss for individuals affected by those crimes. The specific setting for our implementation is based on a framework where mobile phones are used for crime reporting in the developing world. This framework was designed and implemented by Burke and Kayem [7]. The framework captures the limitations we are faced with in a developing country when looking to publish anonymised crime data.

Our implementation assumes that users of the crime database are not able to query for specific records. They are, instead, provided with random samples of the data satisfying k-anonymity and l-diversity constraints. K-anonymisation as first introduced by [49] has become a widely adopted benchmark for data anonymisation. Since k-anonymity is provably NP-hard numerous heuristic algorithms have instead been developed to date for use in various contexts. In a similar manner we implement a genetic algorithm (GA) to provide heuristic solutions for crime data anonymisations that satisfy k-anonymity. We then extend this notion of privacy to l-diversity which seeks to address some of the limitations of k-anonymity.

Users of mobile applications have increased significantly in recent years, driven largely by advances in mobile computing power and networking technologies. Software applications previously considered irrelevant have rapidly become a necessity to everyday life. Whilst the benefits of adopting the mobile phenomenon are significant, users are nevertheless vulnerable to the ex-

tent that increasingly personalised information is being entrusted to a third party. The latter issue has arguably enjoyed less attention and only surfaces in cases of severe security or privacy breaches. The context of mobile crime reporting is however somewhat unique in this respect as users are seemingly more risk averse to privacy loss in this setting. Law enforcement agencies therefore need to ensure that they obtain and retain public confidence in their ability to manage the data about reported incidents appropriately; especially when third parties are granted access to perform analysis on their behalf.

The framework outlined by Burke and Kayem [7] consisted of two modules, namely: (1) a data collection module and (2) a data anonymisation module. Our focus is firmly on the latter where we hope to improve on the manual anonymisation process in their original framework. The majority of our work is therefore dedicated to automating the anonymisation process whilst maximising information utility by using a GA.

1.1 Motivation

A framework and relevant practical considerations for mobile crime reporting in a developing country was introduced by Burke and Kayem [7]. Within that context reports about crime are collected by a law enforcement agency who is the owner of the data. The owner of the data has a duty to maintain the privacy of crime reporters not only from a moral perspective but also from a personal safety perspective.

As mentioned earlier the law enforcement agency might however want to analyse the reported crime data to help facilitate its statutory role of enforcing the law. Crime prevention, avoidance and profiling are amongst the few benefits obtained by crime data analysis. Adequate volume and quality of crime data also assist in allocating scarce resources more efficiently across law enforcement departments within the context of a developing country. One practical example might be to apply Bayesian techniques to assess what police unit should be dispatched to a reported crime scene given only the location and time of the event. One might be able to dispatch the correct unit with highest probability given only those two factors for instance.

Within a developing country it is unlikely that the required expertise will be available within the law enforcement agency itself to conduct statistical analysis of the data. Such analysis may therefore need to be done by a third party which raises privacy concerns. There is a definite need to minimise the probability of disclosing the identity of a crime reporter explicitly or implicitly. Burke and Kayem [7] addressed this by anonymising the data prior to sending it to the third party for analysis. Figure 1.1 shows their implementation. [39] summarises the notion of statistical disclosure and cites works related to it. We will again turn our attention

to the various forms of disclosure and attacks on anonymised data in Chapter 2.

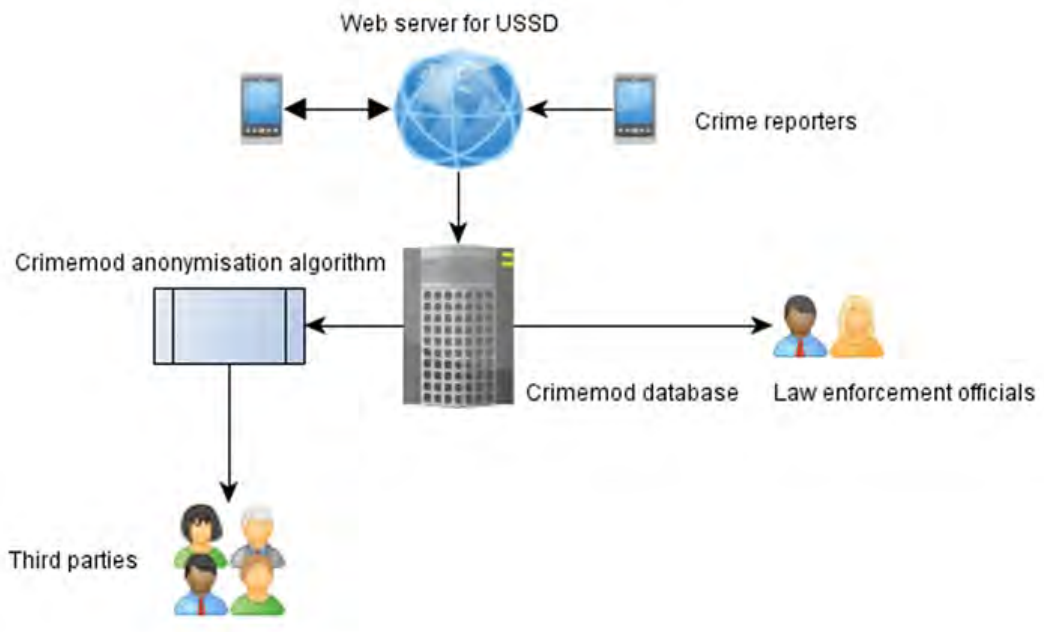


Figure 1.1: Crimemod implementation

We have already noted the inherent trade-off between the benefits derived from crime data analytics and the risk of identity disclosure. It is worth pointing out that there might be additional complexities when outsourcing crime data analysis to a third party. We consider the following scenarios as examples:

- Crime data in particular might be valuable to foreign intelligence agencies. They might be able to identify groups hostile to the ruling government and use this information to attempt to influence local politics
- The third party may be located in another jurisdiction where laws relating to privacy might differ. This could make it difficult to take legal action should any breaches of privacy occur as a result of the data release
- Reported crime data may be released to the third party before any arrests have been made or suspects taken into custody. Where the data has not been adequately or appropriately anonymised, disclosure might assist a suspect in avoiding arrest if he can identify that the crime he committed has been reported. In this instance some co-operation between the third party (who is analysing the anonymised data) and the suspect is required

Whilst the above examples are not exhaustive it demonstrates the need to ensure sufficient anonymity requirements when sharing crime data with a third party.

Challenges and limitations relating to the collection of reported incidents were addressed in Burke and Kayem [7]. We further note that biases may be introduced into the data through this form of data collection. By biases we mean that the reported frequency distribution of crime reports differs from the true frequency distribution. We provide some insights into how this can occur within the framework of Burke and Kayem [7] and how such biases might be addressed during the anonymisation process.

Anonymisation of data requires specific resources and within the context of a developing country numerous limitations need to be contended with. The skills shortage has already been alluded to earlier and is seen as the most significant constraint - however limitations on computational power and time constraints should not be underestimated. The budget of a law enforcement agency in a developing country may not allow for expensive computing devices (such as a server farm for instance) dedicated to the anonymisation of data only. On the other hand anonymised data should be available in a timely manner and not take hours or days before it is available. Third parties will require timeous responses to their requests for anonymised data.

The three limitations above are what we will consider as parameters that control the "resource constrained environment". In this paper we seek to develop an automated scheme for selecting the appropriate level of anonymity to address the skills shortage within a law enforcement agency. Our selection of algorithms and techniques for automated anonymisation are then informed by the associated time and computational complexities.

1.2 Problem

Collected crime reports within the mobile crime reporting framework (MCRF) of [7] are stored in a database. These records need to be anonymised before making it accessible to third parties. Whilst the anonymisation module of [7] provides an algorithm used to anonymise the data before sending it to the third party, the levels of k-anonymity and l-diversity are pre-defined inputs set by a specific security hierarchy. This thesis explores the following questions derived from this setting:

i Can the crime data anonymisation be automated and to what extent is automation possible? For instance, domain generalization hierarchies (DGH) may still need to be specified for attributes unless a suitable automation scheme can be found for this

ii How can data utility be improved from the perspective of third parties if we do not know what analytical tools will be used to extract information?

iii Will the selected automation procedure cope with the dynamic nature of the data source in terms of computational power and time constraints? Crime reports are generated frequently resulting in significant growth in the dataset. Ideally most recent reports should be included as anonymised records in data queries but can this be done in real-time given our constraints?

1.3 Contribution

The overarching theme of this work is to consider the automation of data anonymisation for crime data, specifically within the MCRF setting described in [7]. Related topics are also covered - some extensively - with the aim of supporting and enhancing our main automation goal. Although a unique crime reporting setting was chosen for our implementation, extension of ideas to other fields would seem straightforward.

In a quest to reduce user intervention during anonymisation whilst maximising the utility of the data, this paper therefore contributes the following:

1. We define a crime weighting scheme which facilitates the automation aspect of our work as we no longer need to set the level of k-anonymity or l-diversity based on the end user as was required in [7]

2. A modified loss metric (LM) is incorporated into the GA to ensure that end users can select which quasi-identifiers (QIDs) they seek to prioritise during the anonymisation. This adds flexibility and improves data utility for third party users as they can tailor the anonymised output to better suit their needs

3. We employ a genetic algorithm (GA) approach called "CrimeGenes" (CG) to derive Pareto optimal solutions such that information utility is maximised whilst achieving a required level of k-anonymity. Our GA implementation for k-anonymity is referred to as CG-kanon

4. We propose CG-diverse (based on l-diversity) to address the vulnerabilities of CG-kanon to inference attacks. Our crime severity scheme is modified to suit CG-diverse whilst the QID weighting scheme is retained. Our CG-diverse targets diversity directly rather than implementing it as an extension of k-anonymity. This improves optimisation efficiency and enables us to shorten the algorithm runtime. To the best of our knowledge there has not been a GA implementation of l-diversity based on a local recoding (generalisation) model

5. Our work lastly introduces the notion of biases introduced by the MCRF (the meaning of biases was given in Section 1.1). We look at how such biases may come about and the impact of this on data utility. We propose an adjustment factor to compensate for such biases although we do not extensively discuss experimental results related to this due to the scope of our paper

6. We conduct experiments in Chapter 5 to evaluate our contributions as set out above and show that a GA is indeed practical to implement an automated anonymisation scheme within our resource constrained environment

The implementation of our contributions above are done in a manner that is practical and usable[1]. A web-based interface ensures platform independence from the user's perspective. Server-side implementation is also flexible with ODBC providing the required linkage between the Python-based server and the Java anonymisation engine. Our proposed automation feature is therefore simple to implement on the existing MCRF from [7] as modularity is retained with the only dependency being the original database of crime reports.

1.4 Outline

The rest of this paper is structured as follows: Chapter 2 provides an overview of the literature related to our main contributions as set out above. Chapter 3 gives a succinct description of the design of our automation scheme including technical specifications relevant to our implementation. Chapter 4 and 5 together constitute the largest component as far as the body of this paper is concerned - here we set out the implementation and results respectively. Chapter 6 concludes by presenting a summary of our findings whilst recommending avenues of further research based on this.

[1] An online version of our implementation can be accessed at: http://197.189.230.18:85. Please see Chapter 3 for login details.

Chapter 2

Literature review

Literature related to publishing data whilst retaining privacy has grown substantially in recent years. We present a directed review of related work aimed at supporting our automated genetic algorithm (GA) approach for the MCRF. An exhaustive review of the literature can be seen to be a separate work in itself when looking at [18] for instance.

2.1 Privacy and information

The mere availability of information creates a need for analysis if we believe that some value might be extracted from it. In our specific setting such value is derived form the social benefit of reducing crime through the MCRF. Chapter 1 mentioned that advances in computing capacity of mobile devices and networking technologies has facilitated the integration of electronic devices into everyday life. Society's capacity to generate data has similarly expanded and with it a growing demand to make sense of it all. This is countered by concerns about privacy which have come to the fore in recent years. As a result the data environment is far more complex than a few decades ago and so too the privacy issues related to it. It might be helpful at this early stage to point out that cryptography should not be confused with the field of anonymisation. Cryptography in a most general sense seeks to protect access to data by making it unintelligible. Anonymisation on the other hand seeks to maximise the value of data without disclosing (intentionally or unintentionally) an identity or specific attribute.

Where traditionally, as noted by [39], privacy loss was mainly seen as a function of the actual data released, unintended disclosure incidents such as those by Netflix or the well documented case mentioned by [49] of the Massachusetts Governor who was identified through publicly published medical records resulted in new approaches to privacy preserving data publishing

over the last 20 years. The two core issues that most contemporary research seeks to address can be summarised as follows:

1. **External data.** Uncertainty as to what data is available to third parties and not under direct control of the data publisher complicate risks of disclosure attacks
2. **Information utility.** The inherent trade-off between privacy and information loss is well researched and proven to be NP-hard as mentioned in [10]. This trade-off has prompted us to incorporate the notion of Pareto optimality. The Pareto optimal front comprises the set of points where no more data utility can be achieved for a given level of privacy. Incorporating this requirement into our GA ensures optimal results with respect to information utility post anonymisation. Section 2.5.2.1 considers the Pareto optimal front in more detail

Privacy preserving data publishing (PPDP) is often used in the literature to refer to the confines created for data publishers by the two concepts above. Privacy preserving data mining (PPDM) is a related term and seeks to provide a solution to the problem of PPDP by applying data mining tools. Data mining is generally an information centric activity and hence the adoption of such techniques needs to be applied with the necessary privacy restrictions. Literature on PPDM varies widely but the following two approaches can broadly be distinguished:

1. **Anonymise and data mine.** The most common approach here is to modify the anonymisation algorithms to optimise some data mining metric or methodology. Works by [24],[2],[19] and [44] all utilise data mining tools to measure the performance of their anonymisation schemes. In cases where the anonymisation algorithms are not modified, data mining techniques can still be applied to measure or compare utility of anonymised datasets for data mining purposes post anonymisation.
2. **Mine, then anonymise the released data.** The data owner performs data mining on the private data and then publishes the results in a manner that satisfies anonymity requirements. Alternatively the mining and anonymisation approach may be combined into a single process as was done in [17]. Here an algorithm is derived that induces decision trees that satisfy the k-anonymity requirement. We note however that this approach may not be as practical where granular attributes are present as large decision trees would be required. Furthermore such induced decision trees, geared mainly towards classification, may overgeneralise the data.

The above two categories for PPDM are defined by [10] although that work was published in the context of k-anonymity only. We propose this distinction is appropriate for PPDP in general.

PPDP has become more influential due to problems involving big data requiring automated data analytics. Implementation frameworks such as the widely cited Weka[1] software in [21] has also facilitated experimentation and applications in practice. However, appropriate performance metrics for PPDM are still debatable - we return briefly to consider this in Section 2.4.

Before proceeding with an overview of some specific privacy models, we consider it necessary to distinguish between different kinds of data to which anonymisation methodologies are applied. Traditionally, research on privacy preserving data releases was focussed on the notion of data records contained within a database. Within the MCRF this is still the primary focus as we are dealing with database entries pertaining to crime reports. More recently however two additional areas within the field of data anonymisation have emerged. These are:

- **Transactional data anonymisation** can in a general sense be thought of as the data and data linkages inherent in consumer spending data. The paper by [51] describes this developing area of anonymisation as well as the specific nuances particular to it. Instances where retailers or financial institutions seek advice from a third party consultant serve as a practical example where such techniques may be employed. The application of PPDM in this instance to develop association rule learning is particularly useful. Such techniques are used to learn about consumer preferences and purchasing behaviour whilst retaining anonymity. [58] gives examples of how particular consumers may be at risk of privacy loss unless PPDM is utilised instead of pure data mining. Both [51] and [58] propose algorithms for ensuring PPDP in transaction data

- **Network-based anonymisation** is an area receiving much attention deriving its significance from increased internet and intranet usage over the last decade. The explosion of online social networks has created new ways of applying and thinking about anonymisation as described in [4]. Here nodes and edges within the network serve as inputs to model disclosure risk. The tendency for users to prefer mobile services has also raised concerns about location data and how this may be anonymised without compromising the relevance of content provided to users. The paper by [55] covers such related issues. Concerns about location privacy are relevant in the MCRF where mobile phones are used for reporting although utilising location data is not something we have focussed on in our implementation.

The above two areas are more recent developments in the literature when compared to data contained in relational databases. We summarise our thoughts concerning these two areas of anonymisation as follows:

[1]Available at:http://www.cs.waikato.ac.nz/ml/weka/

i New branches of anonymisation require techniques which differ significantly from traditional database approaches. It might therefore be inappropriate to apply anonymisation algorithms in one field to another due to the unique challenges faced in each

ii The underlying data in these emerging fields can be seen to be behaviour-centric i.e. that behaviour drives and determines what data can be derived and investigated. The traditional database approach is different in the sense that it provides a mapping between an individual and a set number of variables

iii The scope of behaviour-centric data can make it more difficult to properly define the disclosure issues in every instance. There may also be varying degrees of overlap with other fields such as privacy or cryptography depending on the context

It can be seen that what makes transactional and network-based anonymisation different is due to fundamental differences in the underlying data. We note these emerging areas only in passing as our focus in this work remains firmly on relational data as described earlier.

A number of privacy models have been derived over the years to address the need for PPDP of relational data. We provide an overview of such models in the following section before considering a selection of these models in more detail.

2.2 Privacy models

Borrowing terminology from [13] we distinguish between syntactic privacy models and probabilistic privacy models.

2.2.1 Syntactic models

Syntactic models have well-defined data output formats for the anonymised data where privacy traits can often be confirmed by visual inspection of the data. Particular disclosure risks are identified in advance and the anonymisation scheme is devised with a specific set of scenarios in mind. These scenarios take into account amongst other factors the information that might be available to an attacker as well as the syntactic and semantic meaning of the underlying data being generalised. K-anonymity, l-diversity, t-closeness and almost all the variants of each of these fall under syntactic models.

Our focus in this paper will be solely on syntactic models as we apply principles from k-anonymity and l-diversity to a GA implementation. The fact that syntactic models have a

fixed format and well-defined output perhaps make this more tractable from a GA perspective compared to a probabilistic definition of privacy. A GA seeks to create an individual with specific features and characteristics. A syntactic model is analogous in this respect as the anonymised dataset has predefined characteristics as well.

2.2.2 Probabilistic privacy models

Perturbation techniques are possibly the most well-known category of probabilistic privacy models. Here noise is added to values in a dataset to conceal the true values whilst ensuring that statistical properties of the data are invariant to the transformation. [15] points out that there are two categories for perturbation techniques, namely *Input perturbation* and *Output perturbation*. The former randomly modifies the data before answering queries based on the modified data. The latter approach uses the original data to derive query results, these results are then randomly modified. [15] further note that due to privacy limitations discovered in initial perturbation techniques further research in this area was dampened. One such limitation was where the magnitude of noise added to a value was used as the metric for privacy. The data was vulnerable to inference attacks from an adversary who knew the true underlying distributions of the data.

A more recent probabilistic approach which sought to overcome the initial criticisms of perturbation approaches was published by [14]. This was known as differential privacy and has also been the focus of quite substantial research in this field in recent years. Differential privacy in simplistic terms requires that an adversary learn no more from a published dataset when one record (or individual) is present in the dataset compared to if that record (or individual) was removed. The uninformative principle as introduced by [20] and cited in [18] encapsulates this requirement and seeks to limit probabilistic attacks on anonymised data.

Whilst the syntactic and probabilistic models introduced above clearly approach privacy issues for published data from different perspectives they do face some common challenges. For instance both need to contend with the issues surrounding continual publishing of data. Section 2.3.7 of this paper looks at this problem from the perspective of syntactic models. On the other hand, the paper by [16] seeks to address this problem from a differential privacy point of view.

Lastly, we note there have been efforts to create hybrid models which aim to combine the best attributes of both approaches. The probabilistic k-anonymous implementation of [2] and the differential privacy combination with t-closeness by [13] are two such examples. Whilst probabilistic models are better suited to the anonymisation of numerical data, syntactic models

are more appropriate for categorical data. In our MCRF the majority of attributes in the dataset are categorical. For the remainder of this chapter we therefore focus solely on research related to syntactic models and we now turn to explore these further.

2.3 K-anonymity and its extensions

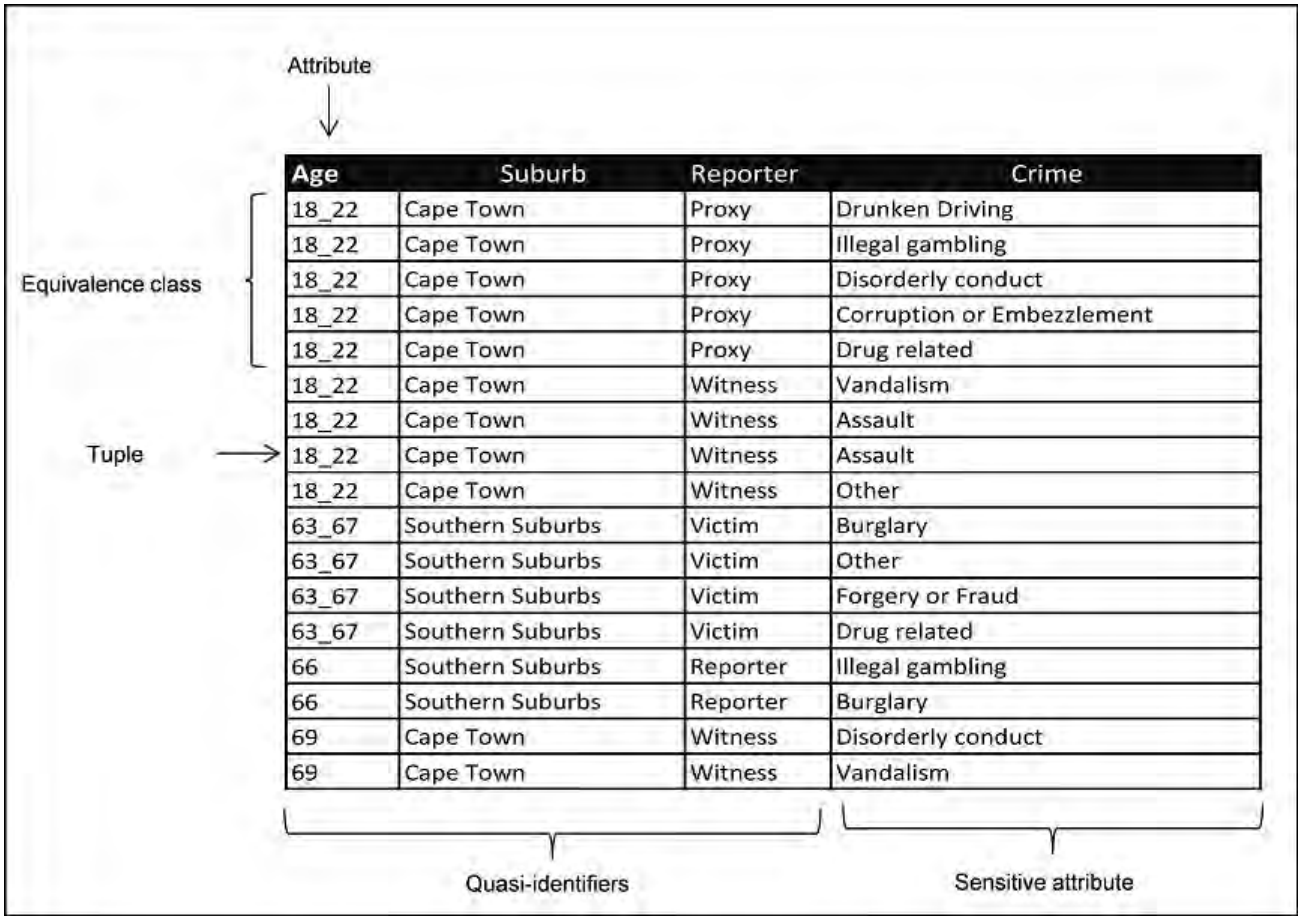
2.3.1 K-anonymity

The k-anonymous model as first introduced by [49] has dominated much of the literature related to privacy preserving data publishing. The following definitions will enable us to formalise the notion of k-anonymity and will be used elsewhere in the paper.

- **Tuples.** Since our data structure is contained within a database a tuple is equivalent to one record entry or a row in the database
- **Attributes.** Within the context of a database an attribute can be interpreted as the columns of a row. The attributes are therefore data fields which when combined create a tuple (or row). The set of attributes may include explicit (name, identity number or surname) or general identifiers (gender, postal code or ethnicity)
- **Quasi-identifiers (QIDs).** These are attributes which independently or when combined can be used to uniquely identify an individual or entity. Most often such identification is done by combining external data with the published dataset
- **Sensitive attribute.** The attribute which, when combined with the QIDs, would result in a disclosure
- **Equivalence class.** [33] define an equivalence class as "a set of records that have the same values for the quasi-identifiers"
- **Disclosure.** The unintended loss or reduction of privacy for an individual or entity resulting from a publication. [33] note that two types of information disclosure can be identified namely, identity disclosure and attribute disclosure. 'Identity disclosure occurs when an individual is linked to a particular record in the released table. Attribute disclosure occurs when new information about some individual is revealed. The released data makes it possible to infer the characteristics of an individual more accurately than it would be possible before the data release'

- **Adversary.** Any entity seeking to extract more information from the released data in order to achieve a disclosure

Figure 2.1 shows how the definitions above relate to a sample anonymised dataset.



Age	Suburb	Reporter	Crime
18_22	Cape Town	Proxy	Drunken Driving
18_22	Cape Town	Proxy	Illegal gambling
18_22	Cape Town	Proxy	Disorderly conduct
18_22	Cape Town	Proxy	Corruption or Embezzlement
18_22	Cape Town	Proxy	Drug related
18_22	Cape Town	Witness	Vandalism
18_22	Cape Town	Witness	Assault
18_22	Cape Town	Witness	Assault
18_22	Cape Town	Witness	Other
63_67	Southern Suburbs	Victim	Burglary
63_67	Southern Suburbs	Victim	Other
63_67	Southern Suburbs	Victim	Forgery or Fraud
63_67	Southern Suburbs	Victim	Drug related
66	Southern Suburbs	Reporter	Illegal gambling
66	Southern Suburbs	Reporter	Burglary
69	Cape Town	Witness	Disorderly conduct
69	Cape Town	Witness	Vandalism

Figure 2.1: An anonymised dataset

Using the above terminology we can informally state that a dataset is said to be k-anonymised if for every entry there exist exactly k-1 records in the anonymised dataset with the same QIDs. Equivalently, the equivalence classes of a k-anonymised dataset contain at least k entries. An adversary is therefore at best able to target a specific individual with probability $\frac{1}{k}$ given the QIDs. The above attack is what the k-anonymous model was specifically designed for - to reduce the probability of successfully linking a record or entry to a specific individual. The fact that this approach still poses inferential vulnerabilities is well-known. For instance in our MCRF we may have an equivalence class consisting of 10 reported crimes where 9 offences relate to murder and only 1 to theft. Where an adversary knows the QIDs of a crime reporter they are therefore able to state with 90% certainty that the reporter reported a murder.

Although the inferential risks in k-anonymised data are addressed to a large extent by l-diversity in Section 2.3.5, a significant body of literature is aimed at improving algorithmic techniques for its implementation. We consider the various adaptations and permutations of this model in what follows.

Two of the most well-known earlier extensions for k-anonymity were published in [30] and [31]. These implementations are commonly known as Incognito and Mondrian respectively. The Datafly algorithm presented in the original k-anonymity work by [49] whilst a first step towards implementing a k-anonymous model gave no guarantees about optimality. The work in [30] sought to address this shortcoming and utilises a bottom-up approach. A lattice is constructed of possible generalisations before a traversal of the lattice seeks to find the minimal generalisation that satisfies the k-anonymity requirement. A number of k-anonymity algorithms are based on different generalisation structures. A brief digression to consider such structures would be helpful at this stage before we pursue our current line of thought in Section 2.3.3 concerning various k-anonymous algorithms.

2.3.2 Domain recoding

A more thorough understanding of generalisation techniques will enable us to better appreciate the differences between anonymisation algorithms in the literature. We use the notion of recoding and generalisation interchangeably. Whilst some of the literature provides extensive formal definitions of recoding domains we restrict our discussion to an informal descriptive approach.

We distinguish as follows between the two recoding models in the literature:

1. **Global or full domain generalisation.** For each attribute, the occurrence of a specific value within that attribute is generalised to the same level. For example if the age 18 is generalised to the range 18_87 for one tuple then all tuples with age attribute of 18 are generalised to 18_87.

2. **Local recoding.** Local recoding models are agnostic of generalisations applied to the same attributes of other tuples in the dataset. For instance one tuple with age attribute of 18 can be generalised to the interval 18_23, the age of another tuple can be left as 18 and another tuple can have age generalised to 18_87.

Although [30] utilised a full domain generalisation model (as in [49]) they point out that local recodings are likely to provide much better information utility.

The definition as presented above may provide an over-simplified view of generalisation models in the literature, the following caveats and additional discussion points therefore need to be kept in mind:

- One can further distinguish between single and multi-dimensional global recoding models. For instance [31] applies a multi-dimensional generalisation approach within a global recoding framework to create anonymised data with more utility. As pointed out by [42] this implies that only tuples within the same equivalence class have the restriction that attributes are generalised to the same level.
- [42] does not, however, use the notion of global versus local recoding even though their proposed cell-based generalisation approach is clearly a local recoding model. They also speak of single dimensional generalisation when referring to full domain generalisations whereas in [30] this terminology is utilised with specific meaning
- The above two points allude to the fact that the terminology used in the literature as pertaining to generalisation models has not been entirely formalised yet, or at least is not being applied consistently
- Lastly, not mentioned above is yet another distinction between hierarchical generalisation and partition-based generalisations. A total ordering is required for the latter which makes it better suited to numerical or continuous attributes. The former is naturally better suited to dealing with categorical data. The paper by [5] was the first to apply partition-based generalisations to k-anonymisation. All attributes in the QIDs are considered as an ordered set and the anonymisation algorithm is framed as a set-enumeration problem. We return to this below when furthering our discussion about k-anonymity extensions.

Our focus on generalisations may look to imply that the only tool for achieving k-anonymity is through recoding the QIDs space to obtain equivalence classes. However as shown in [59] a permutation-based approach is equally viable and they show that it guarantees the same privacy as generalisation-based approaches. Their main motivation is that better answers are obtainable for aggregate queries using permutation-based anonymity than the more common generalisation approach.

As noted by [59] however, their own approach only seeks to break the link between the QIDs and the sensitive attribute and does not address other forms of linking attacks. Furthermore they demonstrate that numerically sensitive attributes need to cover a sufficient range of values within a group to provide sufficient privacy. Otherwise an adversary might be able to infer a value in close proximity to the actual value for individuals in the same group. This they state as

(k, e) anonymity. Here the parameter e requires that the range of distinct sensitive attributes span a range e. For instance if the sensitive attribute is an individual's salary then a value of 5000 for e will require that within the same group the maximum and minimum salary in that group differs by at least 5000.

In a related work by [56] the QIDs are separated from the sensitive attributes into two tables and then connected through a group ID - [56] called their approach "anatomy". However as noted by [59] this approach can be seen to be equivalent to permutation of sensitive attributes in the same group. Anatomy also does not allow for sufficient privacy of numerical sensitive attributes which puts it at a disadvantage in this respect.

We now return again to review other k-anonymous algorithms equipped with an overview of domain generalisations. It should be noted that our focus later in this paper will be on implementing a local recoding model to exploit the benefits shown in [42] from using cell-based generalisations as they refer to it. Figure 2.2 demonstrates the additional information that can be gained through local recoding as compared to global recoding. The example shows a dataset where 2-anonymity was applied. We see that firstly the local recoding approach can produce more equivalence classes and secondly that those equivalence classes contain more granular information.

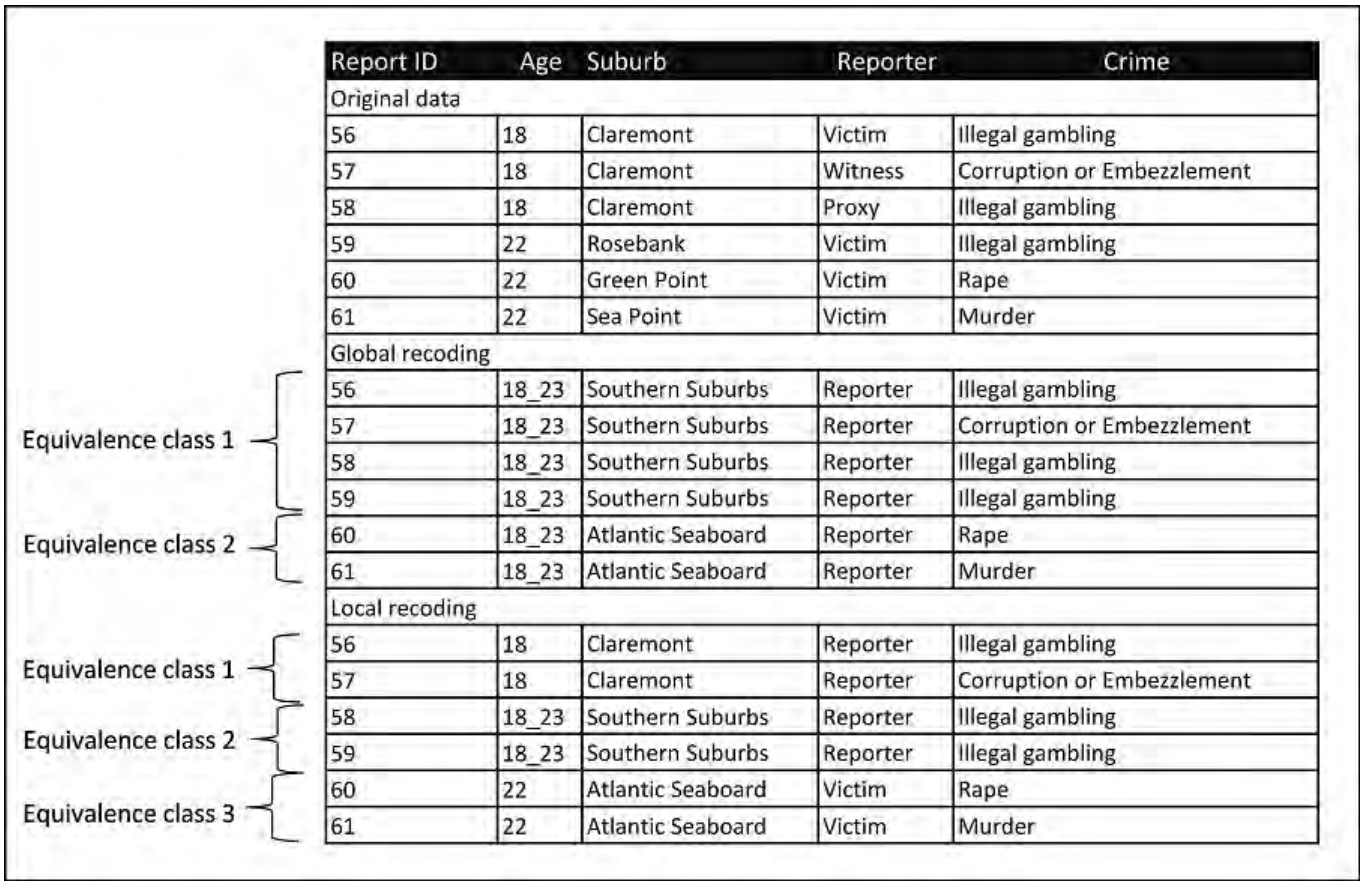
2.3.3 K-anonymity extensions

The overview above relating to domain recodings enables us to consider the Mondrian algorithm in [31] as an improvement on the Incognito algorithm in [30]. Whilst Incognito is focussed on optimality it is exponential in time complexity and therefore quite slow. The aim of [31] was to improve computational speed to achieve reasonable results whilst improving information retention post anonymisation. The former was achieved by adopting a partitioning approach whereas information utility was improved by using multidimensional hierarchies. The top-down greedy algorithm implementation of Mondrian was shown to be $O(nlog(n))$ in time complexity which was a significant improvement. The partitioning, however, requires a total ordering to be established. [8] cites this as one of the drawbacks of Mondrian even though it is efficient.

Other top-down approaches

Whilst [31] above can also be classified as a top-down approach the work by [5] and [19] serve to contrast two quite different implementations of a top-down algorithm. Contrasting these two approaches bring out further insights into aspects of the literature.

1. **K-Optimize.** The K-Optimize algorithm in [5] assumes total ordering of the QIDs and



	Report ID	Age	Suburb	Reporter	Crime
	Original data				
	56	18	Claremont	Victim	Illegal gambling
	57	18	Claremont	Witness	Corruption or Embezzlement
	58	18	Claremont	Proxy	Illegal gambling
	59	22	Rosebank	Victim	Illegal gambling
	60	22	Green Point	Victim	Rape
	61	22	Sea Point	Victim	Murder
	Global recoding				
Equivalence class 1	56	18_23	Southern Suburbs	Reporter	Illegal gambling
Equivalence class 1	57	18_23	Southern Suburbs	Reporter	Corruption or Embezzlement
Equivalence class 1	58	18_23	Southern Suburbs	Reporter	Illegal gambling
Equivalence class 1	59	18_23	Southern Suburbs	Reporter	Illegal gambling
Equivalence class 2	60	18_23	Atlantic Seaboard	Reporter	Rape
Equivalence class 2	61	18_23	Atlantic Seaboard	Reporter	Murder
	Local recoding				
Equivalence class 1	56	18	Claremont	Reporter	Illegal gambling
Equivalence class 1	57	18	Claremont	Reporter	Corruption or Embezzlement
Equivalence class 2	58	18_23	Southern Suburbs	Reporter	Illegal gambling
Equivalence class 2	59	18_23	Southern Suburbs	Reporter	Illegal gambling
Equivalence class 3	60	22	Atlantic Seaboard	Victim	Rape
Equivalence class 3	61	22	Atlantic Seaboard	Victim	Murder

Figure 2.2: Local versus global recoding example

this ordering is used to index the attribute values. The optimal search algorithm is exponential in time and uses the indices associated with the attribute values as part of its set-based manipulations to derive an optimal anonymisation. [5] points out that they sought to remain cost agnostic but provide two example metrics for use in their results, namely a discernibility metric and a classification metric.

The results presented by [5] included an interesting comparison with a hybrid algorithm. This algorithm sought to combine the speed of greedy anonymisation algorithms with the better information utility of stochastic anonymisation algorithms. In a similar fashion to GAs this model had no stopping criterion but continued a stochastic hill-climbing exercise to recursively improve on information loss whilst satisfying the privacy level. Exact further details of this implementation were not given, but could be of value for comparison with our GA implementation as there are few stochastic benchmarks for anonymisation in the literature.

2. **TDR** The TDR algorithm of [19] uses hierarchical generalisation. In [5] the main focus of their implementation was to generate useful anonymised data for classification purposes.

[19] argue that optimal generalisations as presented in [5] are not necessarily optimal for classification. They propose that cost metrics based on the original data and used as a feedback loop in the anonymisation process are less effective for classification. They reference the fact that data mining and machine learning classification techniques are often less effective on the unmodified data even when the unmodified data has the lowest possible cost metric. Other optimal k-anonymisation algorithms seek to minimise error on the training data and thus overfits the classification model. TDR therefore uses "noise" and "redundant structures" to minimise classification error on out-of-sample anonymised data.

Whilst both TDR and K-Optimize start out at the broadest generalisation and refine the granularity until the level of k is reached, their implementations and observations are quite different. Most notably that different opinions exist about cost metrics and that the notion of optimality may often be context specific. This sets the scene for Section 2.4 where we comment on the use of cost metrics in the literature.

More recently machine learning approaches have been applied to the anonymisation process. This has not been restricted to only k-anonymity. The notions of clustering and classification in particular have cultivated a significant body of literature focussed on PPDM.

2.3.3.1 K-anonymity as a clustering problem

[8] were the first to consider k-anonymisation as a clustering problem and provide an algorithm for its implementation which incorporates a classification metric. They show computational complexity of their algorithm is in $O(n^2)$ and therefore in the same order as algorithms seen prior to this. [26] notes however that [8] is sensitive to outliers as it sequentially selects clusters furthest from each other to construct a new cluster. They also suggested that the time complexity can be improved by sorting the QIDs and when this is done their algorithm is faster in $O(n^2/k)$. Their systematic approach finds k- anonymity in a greedy manner and adds records to clusters such that information loss is minimised. They also propose that for a given population a percentage q might be indifferent to having their records published. These records can then be published as is without loss of information. Their algorithm incorporates such records into the anonymised dataset after other records have been anonymised. This reduces total information loss and improves computation speed. This concept could be incorporated in the MCRF through the user interface. One would however need to consider whether users are able to determine their desired privacy prospectively and especially under duress following a crime incident.

An earlier work by [34] also aimed to improve on computation time using a clustering approach. Their algorithm is based on the K-Means algorithm and is implemented as a two stage process. Firstly the clusters are created in a randomised fashion. By creating $\frac{n}{k}$ initial clusters (where n is the number of records and k the level of k-anonymity) the number of clusters (and hence equivalence classes) are maximised. The second process called the adjustment stage modifies the clusters to ensure each cluster does contain at least k entries. The algorithm utilises a randomised approach for cluster creation during the first stage and therefore some clusters may be close to containing k records but not exactly. Both [34] and [26] implement their algorithms using the Adult Census data on the UC Irvine Machine Learning Repository and benchmark their results against the original clustering approach by [8]. Both yield substantial computational time reduction compared to [8] whilst minimising their chosen information loss metrics.

[41] provide another utility driven clustering measure but enable the data publisher to include expert knowledge about data features that might assist (or are critical) to maximizing the anonymised data - they implement this as data constraint rules as part of their algorithm. They provide an example about distinguishing between ages 20 and 21 for publishing anonymised data for alcohol-related research where the drinking age limit is 21. This concept might be incorporated in our setting if crime categories are related to age partitions. However our particular crime classification scheme does not lend itself to this in its current form. Conceptually, however, such data constraint rules could be defined once-off by an expert and incorporated in the automated anonymisation scheme to further maximise data utility for third parties. We do implement a QID weighting scheme aimed at indirectly improving data utility.

The work by [42] was introduced in an earlier section where we considered recoding models. Whilst their approach also applies clustering, more relevant to our resource constrained environment is the notion of a natural domain generalisation hierarchy (NDGH). Without reporting the technicalities involved we cite their results whereby an NDGH is used in an automated fashion to specify the domain generalisation hierarchy. For our automated anonymisation within the MCRF this would be desirable as user input for the generalisation scheme becomes obsolete. Our implementation in Chapter 4 assumes a once-off specification of the generalisation hierarchy. However this may need to be redefined in future as the MCRF grows and develops. An appropriate NDGH may be considered as a possible solution going forward to automate this component of the implementation as well.

2.3.3.2 Classification and k-anonymity

Classification has been mostly applied to assess utility of datasets post anonymisation. Nevertheless one model which directly applies this data mining tool by integrating it into the anonymisation process is the work by [17]. [17] provide a one-step process to create a classification model that also satisfies k-anonymity. They achieve this by training a tree-based classifier (such as the ID3 algorithm or the improved C4.5 algorithm) and modifying the algorithm to check for k-anonymity of the classification tree at each node. Whilst not directly of use in our work this may be considered as a quick and ready approach should the law enforcement agency themselves seek to publish a classification model that meets a specified privacy level.

2.3.3.3 Genetic algorithms and K-anonymity

The first such attempt and by far the most widely referenced work was by [24] where the GENITOR algorithm from [54] was applied to achieve k-anonymity for the Adult dataset. A general loss metric as well as a classification metric was used for experimentation. He used this to infer that anonymisations with more utility are obtainable where the data mining or analytical techniques to be used on the data are known in advance. However the counter argument noted earlier by [19] about the relevance of cost metrics for classification accuracy should be kept in mind. Whilst his GA approach and also loss metrics have some drawbacks, it was nevertheless a pioneering approach given work prior to this. The criticisms one could make are possibly more suited to drawbacks of GAs in general of which complexity and long execution times are most significant. These two limitations have meant that applications of genetic algorithms to k-anonymity have been less widespread than some of the other techniques mentioned above. Works by [40] and [35] are amongst these more rare implementations. They also approach the k-anonymous solution from quite different perspectives (terminology used below relating to GAs will be introduced later when GA implementations are discussed):

- [40] used a GA called data mining privacy by decomposition (DMPD) to partition the dataset into distinct groups such that each satisfies k-anonymity. They note that since each disjoint set does not contain all QIDs it is easier to achieve k-anonymity within each partition. In a similar manner to [31] they avoid the need for a generalisation hierarchy by their partitioning approach. The DMPD also ensures k-anonymity is retained should an adversary rejoin the various partitions. The algorithm is design to optimise classification accuracy and therefore performance is measured by training the C4.5 and Naive Bayes classifiers on the anonymised data.

  At a more technical level the DMPD algorithm seeks to find an optimal feature set par-

titioning. Such feature set partition was introduced by [44]. The work in [40] is therefore essentially based on the GA approach given in [44] but extended to suit k-anonymity with a classification goal. The k-anonymity restriction and classification metric enter the fitness function as part of the multi-objective optimisation process. The SPEA2 algorithm from [60] formed part of DMPD to ensure Pareto optimality for selected solutions. This is the same selector utilised in CrimeGenes and we elaborate on this later. Unlike DMPD, CrimeGenes utilises a local domain recoding without the focus on classification.

- [35] use a clustering technique to segregate the dataset provisionally by grouping tuples with similar QIDs. The grouped tuples are then generalised such that all attributes in the same tuple are generalised to the same level to form an equivalence class. The GA is applied at this later stage to minimise information loss to optimise the generalisations applied to the various equivalence classes. The Michigan approach is used for their GA whereby all chromosomes (equivalence classes in this case) together form the solution set. This is in contrast to the Pittsburgh approach whereby each chromosome in the population represents a solution. CrimeGenes utilises the Pittsburgh approach. Section 2.5.3 discusses this aspect of GA in more detail.

The above demonstrates that the flexibility provided by GAs enable their usage in different forms and at various stages in the anonymisation process. A GA can be applied to either fully take over all aspects of the anonymisation process or it may be introduced selectively (in a modular fashion) to perform a more specific task in the anonymisation process. Specifics of GAs in the literature are reviewed briefly in Section 2.5 when we will revisit some issues alluded to above.

2.3.4 K-anonymity limitations

The limitations of the K-anonymous model have been well documented throughout the literature. We point to the limitations most relevant to our implementation of CrimeGenes rather than aiming to provide an exhaustive review of the topic.

The original work by [49] which introduced the model was accompanied by three limitations namely, unmatched sorting attack, complementary release attack and the temporal attack. The latter two limitations are both related to sequential or repeated releases. This is particularly relevant to our proposal for CrimeGenes where we consider a sampling approach to releasing the data. Section 2.3.7 covers sequential releases as a separate topic owing to its relevance.

Inferential attacks are arguably the most likely risks when only using k-anonymity for PPDP. The problem derives from the fact that the model does not enforce any restriction on the distri-

bution of sensitive attributes within an equivalence class. We may end up with an equivalence class where all sensitive attributes are equal and thereby forgo the privacy k-anonymity seeks to ensure. We see an example of this when looking at equivalence class 1 in Figure 2.3. Although 5-anonymity is achieved for both equivalence classes we can state with high probability that a particular individual with QIDs for equivalence class 1 reported a murder. To address this specific shortcoming l-diversity introduced in the next section was devised by [38].

Age	Suburb	Reporter	Crime
Equivalence class 1			
48_52	City Bowl	Reporter	Murder
48_52	City Bowl	Reporter	Rape
48_52	City Bowl	Reporter	Murder
48_52	City Bowl	Reporter	Murder
48_52	City Bowl	Reporter	Murder
Equivalence class 2			
18_22	Cape Town	Proxy	Drunken Driving
18_22	Cape Town	Proxy	Illegal gambling
18_22	Cape Town	Proxy	Disorderly conduct
18_22	Cape Town	Proxy	Corruption or Embezzlement
18_22	Cape Town	Proxy	Drug related

Figure 2.3: Inference attack on 5-anonymous crime data

The NP-hardness of k-anonymity requires higher order time complexities to ensure optimal results. [9] notes that therefore the complexity for optimal solutions will be exponential in the size of the QID. Admittedly most recent heuristic algorithms can achieve acceptably good results for practical application, but nevertheless from a theoretical stand-point optimality remains computationally expensive. Since no bounds on efficiency and optimality of solution can be placed on heuristic approaches, experimentation has been a crucial aspect of applying such approaches to prove the worth thereof. The metrics used to evaluate each of these may be debatable especially as selection of performance metrics might be subjective and suited to reinforcing the validity of the proposed algorithm.

The curse of dimensionality as shown by [1] means that a large number of QIDs results in unacceptably high information loss. This derives from the fact that with many QIDs the probability that k tuples in the raw data are similar becomes very small. From a practical perspective this implies that accurate specification of a minimal QIDs set is desirable. For most practical cases we have encountered this has not yet been a problem however it clearly does not preclude all instances and one needs to remain cognisant of this limitation.

[12] discusses an often overlooked distinction between k-anonymity as a *method* and k-anonymity as a *policy* to preserve privacy. Whereas the former is well documented and understood we may overlook the context within which we apply this anonymisation technique. This is where the notion of *policy* is important. For example, as [12] points out, from a social perspective we might want to ensure that there is extra protection for elderly and young people when per-

sonal data is published. However because k-anonymisation essentially aims to provide equal protection across tuples, this is not achieved. Another scenario might be where anonymised crime data for a particular region was published and this has an impact on an employer hiring candidates from that area. If a candidate is from a suburb where reports of drug abuse have been particularly high, for example, the employer may be less likely to hire the candidate based on the published data. As a *method* k-anonymity has served to preserve the privacy of offenders, however, this should be weighed against the possible unfair discrimination that innocent parties may suffer subsequently.

The limitations of k-anonymity discussed above can be seen to range from being quite technical to including more softer issues such as policy and social contexts. In spite of these limitations k-anonymity has nevertheless been a useful and widely used tool. In applying it to published data we should however be mindful that some limitations do exist and that its appropriateness for a given scenario needs to be judged accordingly.

2.3.5 l-diversity

l-diversity was proposed by [38] and simply stated requires that the most frequently occurring sensitive attribute in any equivalence class should not occur more frequently than $\frac{1}{l}$. Consider an equivalence class for anonymised data in the MCRF with 10 tuples where "Theft" is the most common crime in that equivalence class. If we require 5-diversity then "Theft" cannot occur more than twice in this equivalence class. Note that another interpretation of this is that there should be at least 5 distinct sensitive values per equivalence class since any given sensitive value cannot occur more than twice.

The above definition is what [38] referred to as distinct l-diversity (where the term l-diversity is used elsewhere in the paper this is the form we are referring to if not stated otherwise). They did however provide two additional stronger forms of diversity, notably entropy l-diversity and recursive l-diversity.

- Entropy l-diversity is based on information entropy, but [38] shows that this may be too restrictive as it implies a minimum aggregate level of entropy across the whole table.
- Recursive (c, l)-diversity ensures that the most frequent sensitive value occurs frequently enough but the most infrequent value not too infrequently.

We note that whilst the three definitions of l-diversity (and specifically distinct l-diversity) may not be complicated the implementation, as with k-anonymity, is crucial so as to obtain

efficient algorithms for achieving l-diversity in reasonable time. Therefore in a similar fashion to k-anonymity there have been multiple implementations of l-diversity in the literature to date. For instance [56] and [59] introduced earlier and based on permutation both allow for l-diversity. In an attempt to minimise the number of passes over the data for an anonymisation algorithm [43] present "instant anonymisation" as an alternative which satisfies both k-anonymity and l-diversity. There have been several other implementations not covered here. However as with k-anonymity there are also shortcomings and these are summarised by [33] as follows:

- **l-diversity may be unnecessary.** The necessity for l-diversity is dependent on the distribution of sensitive attributes in the dataset. Where values in the sensitive attribute have extreme probability mass functions (i.e. some values have very high frequency and others very low frequency) l-diversity may result in significant information loss and may be unnecessary therefore.

- **Skewness.** l-diversity only considers the frequency of specific values in an equivalence class and not the entire distribution for values in that equivalence class. This is arguably the most common limitation of l-diversity and mainly the shortcoming which t-closeness proposed by [33] seeks to address.

- **Semantics.** Whilst an l-diverse equivalence class may contain sufficient variation in sensitive values there is no way to capture the semantics of those sensitive values. An example for the MCRF might be where one equivalence class satisfies 2-diversity say, but only contains reported crimes *Disorderly conduct, Vandalism, Theft.* Where an adversary knows the QIDs for an individual they might gain additional information by learning that a less serious crime was reported. This is in contrast to an equivalence class containing only *Murder, Rape and Arson* where the adversary knows that a serious crime was reported.

To the best of our knowledge l-diversity has not been widely applied using GAs. This may be partly due to the expansion in the search space when introducing another requirement into the fitness function. However [18] note that an l-diverse set is also k-anonymous such that the level of k is equal to l. We might therefore be able to get away with only seeking l-diversity using the GA and thereby obtain a minimum level of k-anonymity as well. This is what our CG-diverse algorithm achieves.

2.3.6 t-closeness

t-closeness addresses the skewness limitation of l-diversity by 'requiring that the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the overall table' as stated in [33]. They propose the Earth Mover's Distance (EMD) as the appropriate metric to determine the similarity of distributions. EMD has been popular as an image retrieval technique such as in [45]. Here histograms of images are compared using EMD before retrieval from a database. However, the EMD arguably lacks the statistical grounding of measures such as the Chi-squared test which [45] also compare the EMD against. [45] provide further comparisons of dissimilarity measures. However, actual results obtained by [45] for these measures as applied to image retrieval have little bearing on what might be suitable given our focus on anonymity in crime data.

There have been attempts to consider the statistical roots of EMD likening it to the Mallows distance such as in [32]. [52] also gives more sound statistical grounding for the EMD, a measure being largely derived from its engineering applications. By applying EMD (traditionally an engineering tool) to computer science, [45] seemed to have sparked interest into refining the calculation of EMD to make it more efficient for use in large data applications - such algorithms have been given by [36],[57] and [25].

[33] refer to future work that might be considered for t-closeness, such as having multiple sensitive attributes. Here one needs to consider two attributes separately and then require that an equivalence class has t-closeness if both attributes have t-closeness separately. Another way is to consider the joint distribution of attributes but in this case EMD cannot be easily extended.

The notion of syntactic and probabilistic privacy models was covered earlier. As pointed out in that section the paper by [13] aims at bridging the divide between these two different philosophical approaches. In particular their work combines k-anonymity for the QIDs and differential privacy for the sensitive attributes to develop a "stochastic" t-closeness. Such hybrid models have been proposed elsewhere but to date not tested as extensively as either of these approaches separately. Further refinements in the literature might make these more suitable for widespread implementation in future.

Applying t-closeness to our work would require another order of complexity beyond l-diversity, which, given our resource constrained environment, would not be feasible under the time constraints. Measures to reduce the runtime for k-anonymity and l-diversity are considered in Chapter 5. Another complication with t-closeness is that we know that a lower t-value is better but how do we evaluate the marginal improvement from a lower t-value. [33] mention that a combination of EMD and the Kullback-Leibler (KL) divergence might be a better approach to

address this question. It can be shown that the KL divergence measures relative information entropy and we return to consider it again when looking at utility metrics in Section 2.4.

2.3.7 Sequential releases

Sequential releases of anonymised data have been somewhat less studied in the literature than the anonymisation techniques themselves. Recently, however, there has been more focus on this issue and since our CrimeGenes implementation is based on a dataset that will continually grow and expand we briefly review literature related to this topic.

One of the first works focused on sequential releases in PPDP was [53]. [18] reviewed developments in this field and [47] grouped their findings into the following categories:

- **Multiple releases.** Here different views or partitions of the data are released at the same time. This is not applicable to our CrimeGenes implementation though
- **Sequential releases.** [46] define this as "several releases of the same table are published over a period of time, where each release contains a different set of the table attributes". This can be seen to be applicable to our work if the MCRF develops and possibly other QIDs are required
- **Continuous data.** Records are inserted and deleted over time and several releases of the same underlying table are published over time

The work by [53], [47] and [46] are all primarily concerned with sequential releases (as defined by [47] above). Here the notion of k-linkability and k-diversity are defined analogous to the definition of k-anonymity and l-diversity respectively. [47] describes these two requirements as follows:

- k-linkability ensures that a combined view of all releases (here tables with different attributes) will not enable an adversary to link any of the QIDs in the tables to less than k distinct sensitive attribute values
- k-diversity requires that an adversary is not able to link any selection of values of the QIDs with any sensitive value with probability greater than $\frac{1}{k}$

[53] was the first to consider issues related to sequential releases, however, their top-down algorithm only focused on instances where there are one prior release of the data. [47] extended

this to consider multiple prior releases. A cell-based generalisation approach was implemented and the addition of tuples was allowed for. According to [46] this was the only study at the time of writing that combined continuous data and sequential releases. There were nevertheless drawbacks to the implementation of [47] most notably that the algorithm depends exponentially on the number of releases and that later releases may suffer from reduced information utility due to restrictions imposed as a result of previous releases. [46] addresses these shortcomings by presenting an algorithm that (1) satisfies a stronger notion of privacy, (2) time complexity is independent of the number of releases and (3) data utility for a release is independent of the release sequence.

We can see from the above discussion that a sequential release in itself requires sophisticated algorithms, which, when combined with a GA approach for anonymisation may require a series of related research works. Due to the more limited scope of our work, which only focuses on continuous data, we therefore opt for a simpler scheme to deal with multiple releases on our crime reported data. Specifically we look at releasing anonymised samples without replacement. This requirement is necessary as due to the randomness of our GA anonymisation and the weighting facility provided to third parties, an unacceptably high risk of disclosure is introduced if previous samples are replaced. We will revisit this issue again later when we discuss our implementation.

2.4 Anonymisation metrics

Earlier in our review of the literature we encountered opposing views about using cost metrics during the anonymisation process. In this section we look at cost and utility metrics in more detail.

2.4.0.1 Cost metrics

[24] was one of the first implementations to utilise a cost metric. There have subsequently been numerous cost metric permutations aimed at improving data utility, including our own. Divergent views exist, however, about whether such metrics are useful. For instance [42] suggest that information loss metrics are questionable and show little correlation between one metric over the other in terms of efficacy of classification afterwards. [31] on the other hand are in favour of cost metrics, in particular pointing to general-purpose cost metrics as a good starting point when the purpose of the data is unknown to the data owner. Similarly [29] note the findings by [42] but show that their own results indicate that lower information loss is related to better classification accuracy using their NSVDist (Non-homogeneous generalization with

Sensitive Value Distributions) algorithm. The paper by [12] notes that numerous works in the literature utilise cost metrics but they express concern that such uniform analysis metrics (UAM) assign the same weighting to attributes when optimising data anonymisation. They question whether this is appropriate since from the perspective of inference attacks, for instance, some attributes may be more useful to an adversary. This is an interesting aspect and our attribute weighting scheme introduced in Section 4.4.3 is closely related to this idea.

A cost metric in itself could serve as a metric to compare anonymisation algorithms. However in much of the literature the use of cost metrics have been restricted to the optimisation process itself and not to assess the appropriateness of the end result. The variety of cost metric definitions also make it difficult to find a standard means for assessing the relationship between cost metrics and data utility. A large number of works select classification as the appropriate data utility measure.

2.4.0.2 Classification

Classification accuracy has arguably been the most popular measure of data utility post anonymisation. Whilst there are definitely instances where this is desirable there are numerous uses of data post anonymisation and we should not be confined to classification accuracy as our only benchmark. Poor classification accuracy may be a symptom of the underlying data and not the anonymisation process itself. One should also distinguish between classification of anonymised data and classification of data that is regenerated from an anonymised dataset. The latter does not induce spurious accuracy whereas the former could. The work by [22] evaluates various uses of anonymised datasets mainly from a classification perspective. Since the focus of our work is not on optimising anonymised data for classification we do not elaborate further on this utility metric. Instead we turn our attention to arguably a more universal metric which would lend itself better to being agnostic of data usage post anonymisation.

2.4.0.3 Kullman-Leibler divergence

The KL divergence was briefly introduced earlier during our discussion of t-closeness. We pointed out in that section that the KL divergence provides a measure of information entropy. We formalise this notion here from a statistical perspective by referring to the work of [27]. They show the relationship between KL divergence and maximum likelihood. Maximum likelihood is well documented in the statistical literature and widely used for statistical inference. This observation would suggest the KL-divergence provides a more standard measure for assessing anonymised datasets from a statistical perspective which is also more independent of its usage.

The KL divergence considers the original data as coming from some multivariate distribution without making any assumption about the specific distribution. It then measures the "distance" or "closeness" of the anonymised data to that original distribution. Some adjustments and assumptions are needed in the anonymised data to make it suitable for the KL measure. We cover these later in Section 4.10. L_pNorm is yet another metric used for utility however we do not cover it here.

Whilst there are clearly divergent views about utility metrics for anonymised data we propose that the overarching principle should remain the same. The chosen metric should reflect the purpose of the anonymised data. Utility metrics alone should not be totally relied on without looking at qualitative aspects of the data as well. Summary statistics may conceal privacy issues which could easily be detected through inspection of the data. As k-anonymity and its various extensions never guarantee total privacy, utility metrics should always be weighed against the opposing privacy risks.

Our review of the literature thus far indicates that the anonymisation of data is a multidimensional problem where factors can often be diametrically opposed. This complexity lends itself to the use of genetic algorithms (GAs). This is because a GA approach enables us to be less concerned about how the optimisation takes place and more with what the results should look like. The next section concludes our literature review and considers aspects of GAs most relevant to our work.

2.5 Genetic algorithms

The use of genetic algorithms in anonymisation of data have been more limited than other techniques. Works applying GAs and referenced in this paper include [24],[44] and [35]. This section covers concepts related to GAs relevant to our implementation and whilst not focused entirely on anonymisation are nevertheless required to appreciate specifics of our proposed solution.

2.5.1 Preliminaries

Genetic algorithms could be seen as a branch of the broader machine learning literature. GAs have, however, only recently become more feasible for practical real-time applications as computing power has increased and become more affordable. An example of this might be to compare the first GA for anonymisation presented by [24] which took 18 hours to run in 2002 to our CrimeGenes implementation producing useful results after 30 minutes. Admittedly the

differences between the two implementations do not warrant a direct comparison, but nevertheless this should give an idea about the order of magnitude by which runtimes have improved over the last decade.

Genetic algorithms are inspired by the observation that physical differences between species of the same kind in nature are attributable to their genetic make-up. Often very small differences at the chromosome level can lead to significant differences in appearance. In an analogous manner we seed our GA with chromosomes that are suited to the particular problem at hand. Each chromosome provides a solution set by applying a slightly different approach to solving the problem. We generate a large number of these chromosomes randomly (within limits for the particular kind of problem) to ensure that a sufficiently large variety of possible solutions are combined into the gene pool. The chromosomes can be represented by binary strings (as was done for earliest GAs) or by text strings as done in our CrimeGenes implementation.

Once the required number of randomly generated chromosomes are available, systematic and iterative processes are employed to generate new chromosomes which better resemble the desired solution. Operators such as selection, mutation and cross-over are typically employed to achieve this. Chapter 4 of our implementation will expand on particulars of a GA in more detail. For the moment we consider a specific class of GA known as evolutionary algorithms (EAs) which are better suited to solving multi-objective problems.

2.5.2 Evolutionary algorithms

Evolutionary algorithms are one particular class of genetic algorithms well-suited to solving multi-objective problems. They have successfully been applied in a variety of fields accompanied by a large and growing body of literature. We therefore present a concise overview of the most relevant concepts for our implementation below. We use the terms *evolutionary algorithm* and *genetic algorithm* interchangeably going forward as the former is a subclass of the latter.

2.5.2.1 Pareto optimality

Pareto optimality is based on the notion that no single optimal solution exists to a given problem but rather that the solution space consists of a set of optimal points. As defined in [28] "a Pareto optimal solution cannot be improved with respect to any objective without worsening at least one other objective". Pareto optimal solutions are applicable where multiple but conflicting factors determine the solution. It is therefore not possible to optimise the outcome with respect to all inputs as inherently a trade-off exists to achieve the result. Pareto optimality is well-studied in Economics for that reason. In our setting the trade-off for anonymisation exists

between information utility and privacy. This we seek to solve with the GA approach. The other trade-off to consider is the resource constrained environment within which we operate where time, computational capacity and a skills shortage needs to be considered. This latter less quantifiable problem relating to constrained resources requires judgement more than numerical analysis and we consider such related issues where appropriate throughout the paper.

Returning to the Pareto principle we point out that EAs for multi-objective problems seek to derive an approximation to the Pareto optimal set, more specifically the Pareto optimal front. Such solutions are commonly referred to as non-dominated solutions. However it is worth pointing out that EAs for multi-objective problems are quite different from EAs for single dimensional problems. In particular there are two major differences as highlighted by [11]:

- Multi-objective EAs require a selection mechanism based on Pareto optimality. Using GA terminology, the selection mechanism refers to the process of selecting only the fittest individuals to breed the next generation of results. The Opt4J framework of [37] which we implemented requires one to choose such a selector. As mentioned in Chapter 5 this is where we opted for the SPEA2 algorithm as proposed by [60]
- A mechanism to ensure sufficient variation in the population is required for multi-objective EAs to avoid the solution converging to one point in the solution space. Convergence to a single point is desirable for single dimensional EAs but not multi-objective EAs which often need to derive a set of optimal points

The SPEA2 (Strength Pareto Evolutionary Algorithm) referred to above and the NSGA2 (Non-dominated Sorting Genetic Algorithm) are two well-known algorithms designed to give Pareto optimal results.

2.5.3 Pittsburgh vs Michigan approach

The Pittsburgh and Michigan approach refers to a design feature of GAs and relates to how the chromosomes are constructed for modelling a solution. The difference between these approaches can be summarised as follows:

- **Pittsburgh.** Here each chromosome represents an entire solution to the problem. In our CrimeGenes implementation one chromosome is therefore one possible anonymisation of the entire dataset
- **Michigan.** The Michigan approach is different in that all chromosomes combined represent a single solution for a problem. This approach would require the dataset to be

broken up such that different chromosomes model different tuples in the dataset. The solution is then created by combining all chromosomes.

CrimeGenes implements a Pittsburgh approach. The work by [35] uses a Michigan approach by constructing chromosomes out of partitions used during a clustering pre-processing step. As noted in [35] care needs to be taken since the index of each tuple in the dataset should occur exactly once only when dealing with anonymisation. Therefore applying a mutation operator on the chromosomes in this setting would invalidate the generalisation. We are therefore restricted to cross-over operations only.

Work in other areas such as [23] have specifically explored the merits of using either approach, albeit in a very different setting. From a design perspective the Pittsburgh approach is simpler to construct. Generally, however, we note that the Pittsburgh approach requires more memory and computational capacity especially as the population size is increased. Consider a dataset with 100 records where the GA has a population size of 100. The algorithm therefore needs to handle 100^2 genes whereas with a Michigan approach there are still 100 genes irrespective of the population size. With the Pittsburgh approach it is however possible to introduce more variation into the optimisation process and therefore convergence to the Pareto optimal front might be quicker. Where computing capacity and memory capacity enables such implementation this might be desirable. One could of course reduce the population size in a Pittsburgh setting where memory constraints exists, however, this negates the benefits to be derived from a larger gene pool during optimisation. A redesign of the problem might be preferred in that case in order to implement a Michigan-based approach.

In the CrimeGenes setup our sampling approach reduces the gene pool significantly and therefore problems with memory capacity were avoided. However, one should remain mindful of such restrictions when designing the GA and we again revisit this issue briefly in Chapter 5.

2.5.4 Stopping criterion

A number of works in the GA literature are dedicated to deriving the appropriate stopping criterion for the algorithm. The meta-heuristic nature of EAs and flexibility provided for specifying and solving complex multivariate problems comes with the drawback that placing a limit on the runtime is less well-defined. The paper by [3], although less contemporary, demonstrates attempts to specify upper bounds for GA computation times to facilitate their practical implementation. One reason why a more standard approach for terminating the GA has not been adopted might be attributed to the fact that there are different priorities for different implementations of a GA. In our case for instance the practical considerations for the

resource constrained environment dictates where our focus should lie. The algorithm can not consume inappropriate amounts of time and computing power. However, in a purely research-based setting theoretical aspects of a problem override the need to achieve immediate results and the algorithm can run for hours or days. Nevertheless we briefly state possible approaches for terminating an EA and our thoughts on how relevant these are for CrimeGenes.

Possible termination criterion might be one (or a combination) of the following:

- **Maximum number of iterations (generations)**
- **Limit on execution time.** Here a distinction is required between the notion of time from a user's perspective (user-defined time) and CPU time (CPU time). A computing device may be interrupted when other resources require CPU time. This delays the GA and user-defined time is increased
- **No new pareto optimal solutions.** The algorithm terminates if there are no new non-dominated solutions in the archive after a set number of iterations or after a pre-determined user-defined time
- **Variance of fitness values.** This approach was given in [6] and the GA stops once the variance of non-dominated solutions (i.e. the values for their fitness functions) is below a specified value. The difficulty here, however, is in setting the appropriate level given variations in the data going forward as well as user preferences in terms of attribute weightings introduced later in our work

Other termination criterion are covered in the literature, however we view the above as most relevant to our implementation. CrimeGenes sets a limit on the user's perception of time before terminating the optimisation. A note on CPU time is perhaps appropriate in this regard. As the number of concurrent third party users requesting data increases, a set limit based on user-defined time will reduce the quality of results produced. This should be kept in mind and monitored during and post implementation. In such cases a more appropriate stopping criteria might be to set a limit on the number of generations (iterations) through an experimental process as done for the time limit in CG.

This concludes our review of topics in the research literature related to our work. The following section introduces our implementation by detailing its design and layout.

Chapter 3

Design and Specification

The contribution of this paper focuses on a subset of the original MCRF introduced by [7]. The schematic in Figure 3.1 demonstrates how our implementation known as "CrimeGenes"(CG) relates to the original MCRF. [7] focused on a broad range of issues ranging from mobile users doing the crime reporting to considering privacy issues for such data released to third parties. Interfaces between the framework components were also set out in some detail by [7].

We see from Figure 3.1 that our focus with CG is directed at the anonymisation aspect of the MCRF. In a modular fashion, we seek to provide an automated anonymisation scheme whereby data utility is maximised through a genetic algorithm(GA).

3.1 System requirements

Whereas the MCRF introduced and focused on the mobile aspect of the framework, we seek to extend the anonymisation capability of the original framework and focus on making the data accessible to end users for information or analysis purposes. In light of this the following design considerations are taken into account:

1. End users should be able to access data (obtain output) through a single integrated platform in a secure manner that is user-friendly. Here users include both law enforcement officials and external third parties

2. The system should provide data output with automated anonymity based on minimal user intervention. It is worth remembering that data anonymisation is a means to preserve privacy. In this respect our requirement here for automated anonymisation can be seen to be closely related to the notion of privacy by design (PbD). We do not elaborate further

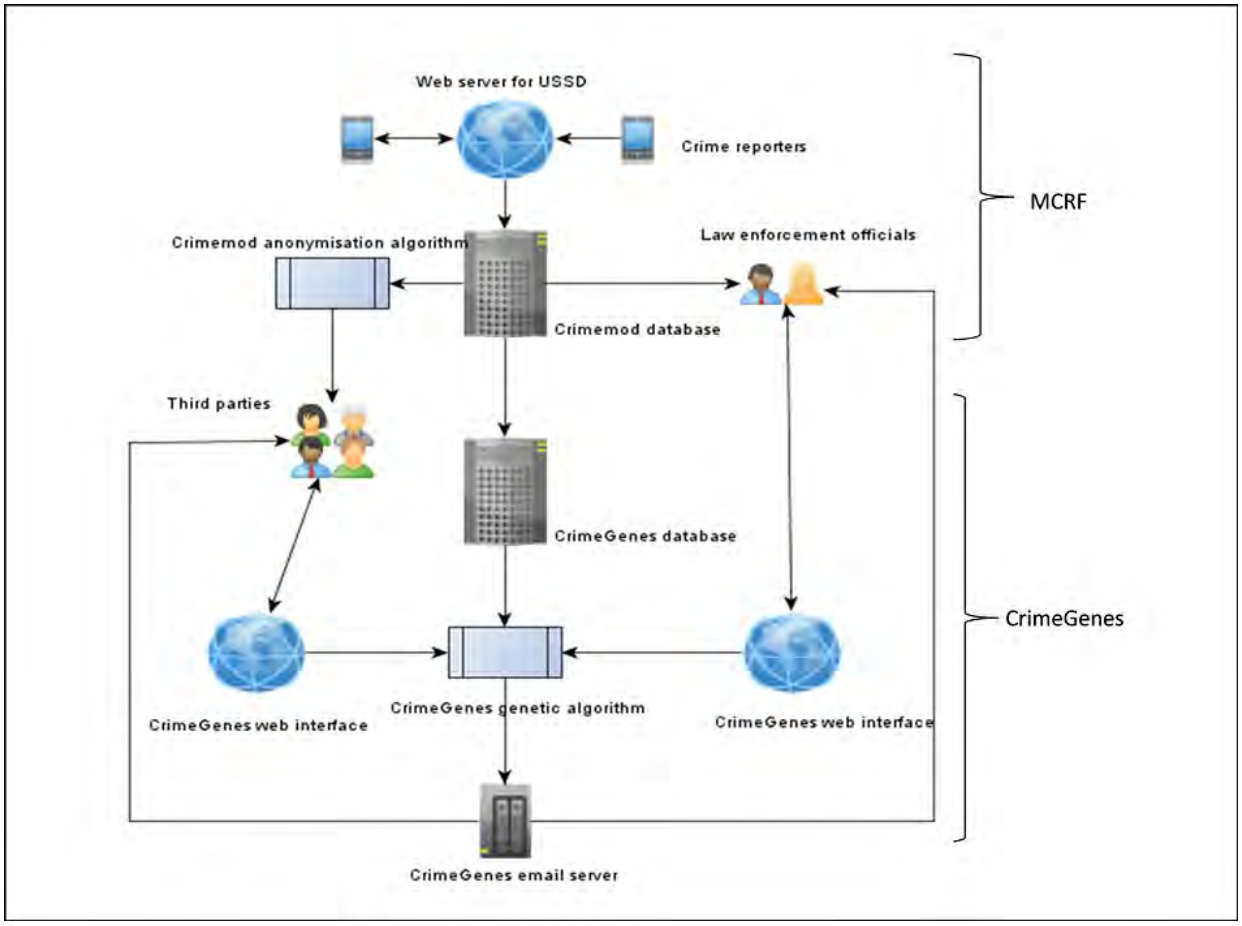


Figure 3.1: CrimeGenes anonymisation

on PbD in this paper, however the reader is referred to [48] for a concise introduction to this field in the privacy literature

3. The data returned should reflect most recent crime reports as far as possible whilst still adhering to anonymity constraints

4. Scalability is important to ensure that growth in volume of data (crime reports) and data users (third parties and law enforcement staff) can be accommodated

5. The CG implementation should be modular and be easily integrated into the existing MCRF implementation or another crime reporting system if needed

3.2 Module components

We implement our solution using a web interface whereby a user requesting anonymised data obtains such records through a web browser. A web-based system has the advantage of be-

ing more accessible provided adequate internet connectivity exists. There are however also downsides of which security is possibly the main concern.

To provide an integrated interface for all data users we propose a design combining database and web technologies. There are three main modules: A database module, the anonymisation engine and the web server. This section provides a brief outline of each module.

3.2.1 Database

We selected PostgreSQL (version 9.1) as our database management system. PostgreSQL is open source software and widely used in a variety of applications. A large development community has ensured that seamless integration with Java and Python is possible - all other major programming languages are also supported. The original MCRF implemented a MySQL database. It should be noted that our implementation utilises a separate database management system (DBMS) solely dedicated to the management of the CG data. To facilitate this we require a form of log-shipping to synchronise new reports from the original MCRF database to the CG database.

An alternative approach would be to use the original MCRF database and only add additional tables to handle requirements for CG. Whilst an additional database requires twice the storage capacity, this has to be weighed against the modularity retained through our current implementation. We have chosen the latter approach for the time being due to the flexibility it provides going forward should any modification be required.

3.2.2 Anonymisation engine

The anonymisation engine implements the GA which optimises the level of information for a given degree of anonymity. Our definitions for anonymity are introduced in Section 4.6.1 and Section 4.4.3 as k-anonymity and l-diversity. We call the respective algorithms CrimeGenes-kanon (CG-kanon) and CrimeGenes-diverse (CG-diverse).

The anonymisation engine is implemented in Java using the Opt4J framework from [37]. Java was selected for this module to facilitate parallel processing where multi-core or multiple processors are available. The engine uses multi-objective optimisation algorithms to obtain a Pareto optimal set of solutions. In particular we use the SPEA2 algorithm developed by [60] as a selector for the GAs of both CG-kanon and CG-diverse. The need for a selector was discussed in Section 2.5.2.1. Our preliminary testing did not suggest a significant difference between SPEA2 and NSGA2. We therefore opted for SPEA2 as it had already been applied to an anonymisation

problem by [44].

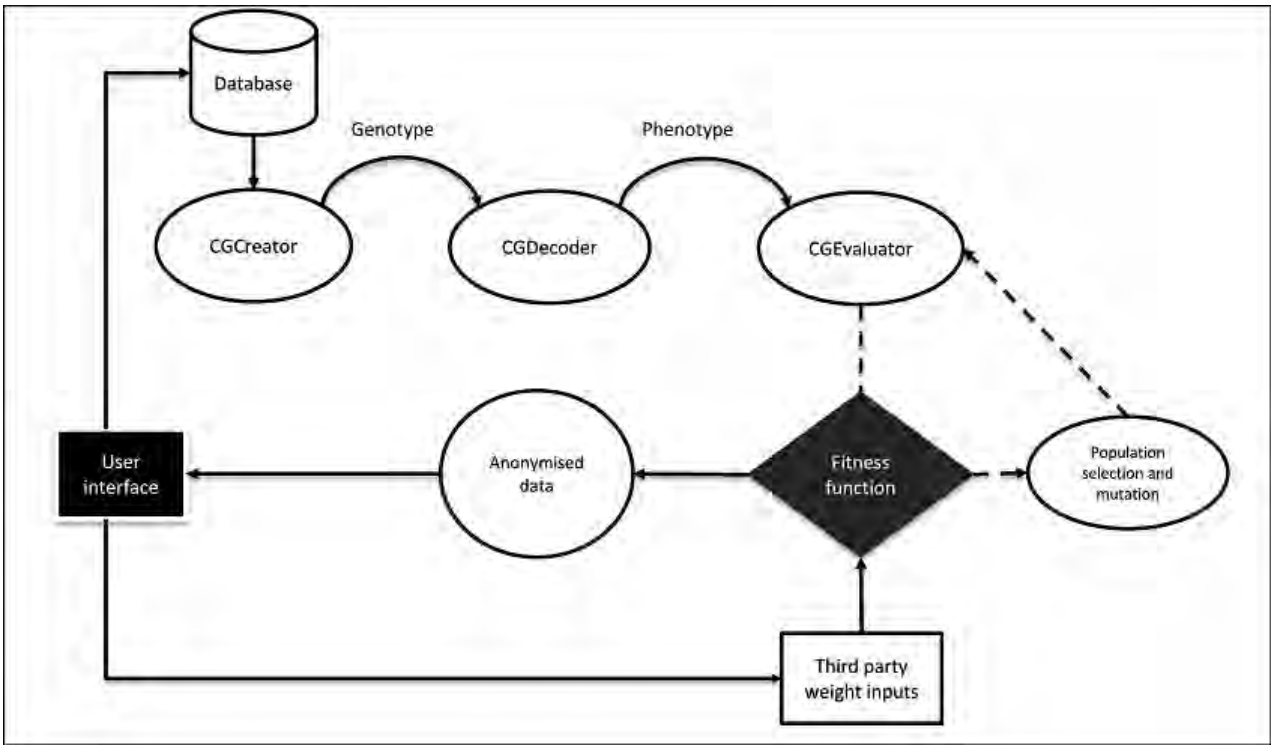


Figure 3.2: CrimeGenes anonymisation algorithm

3.2.3 Web server

3.2.3.1 Server side

The web server is written in Python and based on the Flask web framework. Gevent which is a coroutine-based network library is included to handle high I/O requirements should the number of users or requests grow over time. Gevent does not constitute parallel processing but instead efficiently manages queues to increase performance where I/O demands are high. This is typical for a web server where database queries need to be managed and web pages retrieved for users.

To provide dynamic content for web pages web sockets using the socketIO protocol are utilised. Interactive data is transferred between server and client in JSON format. Once a user has confirmed an anonymisation request the web server initiates the anonymisation engine with the requested parameters and the appropriate data sample. The anonymised data is saved in CSV (comma-separated values) format and emailed to the user once the runtime for the anonymisation engine is completed. Whilst our experimental implementation did not use encryption, a

private/public key encryption scheme is proposed to ensure data integrity and security during transfer. The file containing the anonymised data would be encrypted by the law enforcement agency with the public key of the third party before the file is emailed. The third party can then decrypt the file with their own private key.

3.2.3.2 Client side

Users of the interface are assumed to be computer literate to the extent that they can interact with a graphical user interface - therefore a command line interface is not appropriate. These two assumptions are aligned with our setting of a resource constrained environment - users are more likely to prefer a graphical user interface.

The user interface is created using client side technologies including HTML, JavaScript and Angular JS. From a user's perspective any suitable web browser would suffice to access the system[1]. Secure sockets layers (SSL) is proposed for all web traffic. Access rights will be set by user group. Authentication to access the online system utilises a combination of password and one-time-pin (OTP) made available per user on a daily basis. [2]

The requirement for an OTP was thought necessary as the system is web-based and therefore an extra level of security seemed appropriate. We may however consider two-factor authentication (TFA) as yet a further security enhancement. TFA typically combines some prior knowledge of the user (e.g a password) with a physical device owned by the user (e.g. an electronic token or a cellphone) during the authentication process. This makes unauthorised access more difficult as an adversary requires access to both the password and the physical device which use should be restricted to the intended user only. In our scheme the OTP is sent by email to the user on a daily basis and the OTP is valid for 24 hours. Slight modification of our proposed scheme would satisfy the requirements for TFA if, for instance, we send an OTP for each requested login to a user's cellphone instead. Whether this is practical from a third party's perspective might be debatable. For instance, managing registered cellphone numbers for staff at different third parties might become difficult due to staff turnover at such third parties. From this perspective using only a daily OTP and company email addresses at a third party might be easier, but the lower level of security needs to be weighed against this.

For the purposes of our research the exact details of this aspect are perhaps beyond the scope of this paper. However, if implemented in practice issues around security from the client side

[1] A demo implementation can be found at http://197.189.230.18:85

[2] The online demo version uses OpenID for user authentication and not the authentication scheme above. This makes it more accessible for interested readers. Please use the following OpenId from Yahoo! Username: cg_guest_user@yahoo.com. Password: cg_d3mo

should be explored further.

Chapter 4

Implementation

4.1 Crime data

We begin this section with a description of the data and salient characteristics pertaining to crime data. This will set the scene for Section 4.3 where we look at appropriate generalisation hierarchies for this data. We then introduce the various components of our implementation known as "CrimeGenes" (CG) starting with a modified loss metric in Section 4.4.

4.1.1 Crime data in general

In order to devise an anonymisation scheme for crime data we require a classification system for offences. Categorisation of crime data can vary significantly due to the categorisation of offences being driven by several factors which could include:

- Legal jurisdictions - the classification of offences may differ significantly across states, provinces and countries
- The time period over which crime data is collected and analysed - consider the rapidly evolving field of cyber crime. A few decades ago such classification hardly existed.

In line with [7] we utilise the FBI classification scheme to derive categories for our crime data. This also facilitates comparison of our automation scheme with the original MCRF.

The table in Figure 4.1 shows a sample extract from our generated crime data which consisted of 10000 records. We kept the same attributes for our data as was specified in [7] for the original MCRF but generated our own data to ensure more records can be easily added if required. The

attribute distributions are shown in Figure 4.2. The frequency distribution of reported crime types is exactly the same as in the original MCRF.

ID	Firstname	Lastname	Gender	Age	Address	Suburb	Region	Cell number	Reporter type	Occurrence	Time	Crime type
1	Heather	Cornish	F	30	Franklin Court	Tamboerskloof	City Bowl	0893068604	Victim	Bishopscourt	9	Drug related
2	Matt	Brown	M	49	Route 30	Maitland	Northern Suburbs	0866103857	Victim	Monte Vista	14	Theft
3	Anna	James	F	63	Durham Road	Schotse Kloof (Malay Quarter)	City Bowl	0898842854	Victim	University Estate	15	Corruption or Embezzlement
4	Joe	Simpson	M	26	Virginia Street	Edgemead	Northern Suburbs	0874702036	Victim	Kenilworth	8.5	Illegal gambling
5	Virginia	Dyer	F	67	Overlook Circle	Kreupelbosch	Southern Suburbs	0812064642	Victim	Oranjezicht	16	Illegal gambling
6	Andrea	Ellison	F	62	Front Street North	Bishopscourt	Southern Suburbs	0812559236	Witness	Maitland	12.5	Drunken Driving
7	Alan	Butler	M	23	2nd Avenue	Tamboerskloof	City Bowl	0826338733	Victim	Observatory	6.5	Burglary

Figure 4.1: Sample of generated crime data

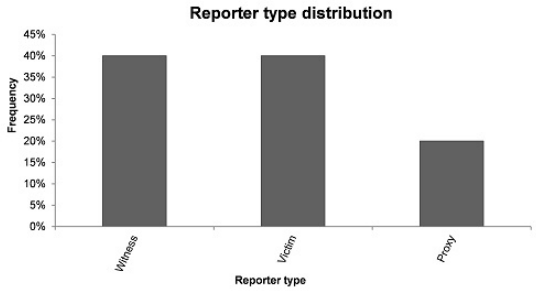
Before proceeding we introduce the following notation related to our dataset to be used later in our discussion:

- O_c is the reported number of offences in category c where $c \in C$
- C denotes the categories of offences
- O represents the total number of observed reports across all categories

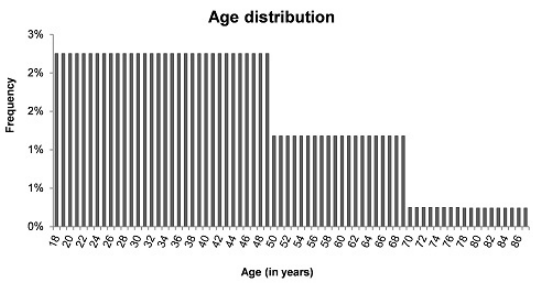
4.1.1.1 Generation of dataset

Figure 4.2 shows our assumed distributions in generating our crime data. Brief comments relating to these distributions are given below.

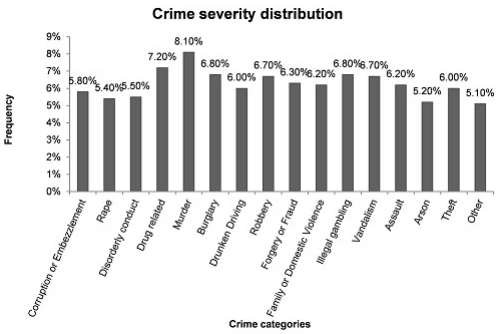
The frequency distribution of crime categories in Figure 4.2(c) was taken from the original MCRF. From this distribution crimes were generated and randomly assigned to various tuples. The fact that reported crimes were randomly assigned to tuples meant that we did not infer any correlation between crime categories and attributes of the crime reporters. The implications of this are that we should not expect high classification accuracy for the raw data as well as anonymised data. This was done specifically to avoid spurious accuracy by not generating data suited to a specific performance metric. Instead our generated data could be viewed as a worse-case scenario and this is preferable to measure the merits of our anonymisation approach. Our selected generalisation hierarchy in Section 4.3 is also specified with this is mind. The hierarchies for *Age* and *Suburb* are both very granular which results in a large number of possible generalisations. Information loss and KL-divergence can therefore be expected to be quite large.



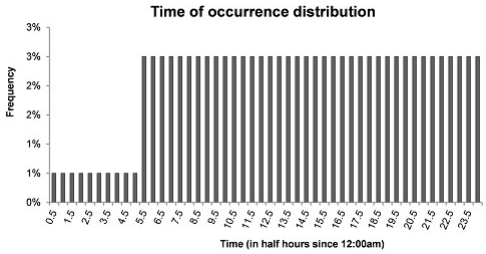
(a) Reporters



(b) Age



(c) Reported crime categories



(d) Time of crime occurrence

Figure 4.2: Attribute distribution assumptions

Our age distribution in Figure 4.2(b) implied that younger individuals were more likely to utilise the MCRF. This is because younger individuals are expected to be more comfortable with mobile applications when compared to older individuals. The reporter type distribution in Figure 4.2(a) assumed that the likelihood of someone acting as a proxy was lower compared to a victim or witness reporting a crime. The reasoning here was that in general reporters should have confidence in the system to preserve their privacy. Therefore, only in severe cases of trauma following a crime, or where there was extreme fear of privacy loss would a proxy report be made. Lastly, we assumed that in general crimes were less likely to occur and be observed during the very early hours of the morning as shown in Figure 4.2(d). By this we mean that both occurrence and observance had to take place. Some crimes are possibly more likely to occur at night (especially serious crimes), however, fewer witnesses are awake to observe and report crimes. The distribution for the time of occurrence was included for completeness only in order to be consistent with the attributes of the original MCRF. We did, however, not include this as a QID when implementing our anonymisation algorithms.

4.1.2 Crime reporting specifics

This brief section introduces points unique to crime reporting which should be kept in mind for when we cover anonymisation of crime data and the subsequent analytics to be performed on this data by third parties.

- Reporting issues. Reports may be misclassified for closely related crimes if reporters confuse two categories. There may also be multiple reports about the same incident. This needs to be taken into account when publishing the data as observations will be skewed
- Distribution of offences. The distribution of reported crimes is likely to be skewed towards less serious crimes. The combined frequency of less severe crimes is likely to be higher than for serious crimes, i.e. we would expect a larger proportion of reported crimes to consist of less serious crimes
- Moral hazard. The fear of being identified may result in reporters misrepresenting personal information whilst only stating facts relating to the crime accurately. Some reporters may go further and misrepresent facts about the crime itself if they believe that revealing certain facts about the incident may put them or their relatives at risk. Unfortunately there is little one can do to address this issue in practice
- Selection bias. There might be various forms of selection bias due to income, education or ethnicity. For example, the fact that a mobile phone is required to report a crime introduces a selection bias whereby specific crimes may be reported more frequently than others due to the reporter belonging to a specific income band. For instance, we might expect a relatively higher incidence of petty crimes reported from residents living in a higher income suburb than lower income suburbs even though the actual frequency of petty crimes might be higher for lower income areas. The impact of perceived severity and propensity to report a crime skews the data relative to the true distribution of crimes.

  These last two points might lead us to consider means of countering such biases through our anonymisation before making the data available to third parties for analysis. The additional information available to law enforcement from prior experience or data gathered might assist in deriving such adjustment factors. These adjustment factors might be used to optimise information utility. This notion of an adjustment factor appears quite lucrative and we will consider this in Section 4.8 as an extension of our core contribution.

4.2 Privacy

[7] stated that all attributes for crime data are considered sensitive and presented algorithms to deal with multiple sensitive attributes. The underlying nature of the reported data means that users of the MCRF are acutely aware of the personal and privacy risks associated with it. Use of automated location information through triangulation for example or fear of corrupt law enforcement officials might deter the public from reporting crimes if they feel their privacy is at risk.

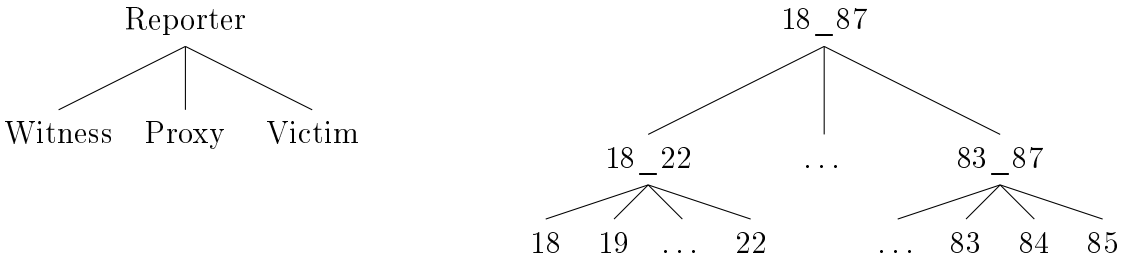
The notion of privacy can take many forms. For the purposes of this paper we seek to preserve privacy by only using k-anonymity and l-diversity. As covered in Chapter 2, we are well aware of the limitations implied by this. However besides the usual challenges of retaining both data utility and privacy we are faced with additional trade-offs in our resource constrained environment where time and computational complexity also plays a role. Hence whilst additional privacy measures could be introduced for an automated genetic algorithm(GA) implementation in the MCRF we do not cover those in this paper.

4.3 Specifying the domain generalisation hierarchy

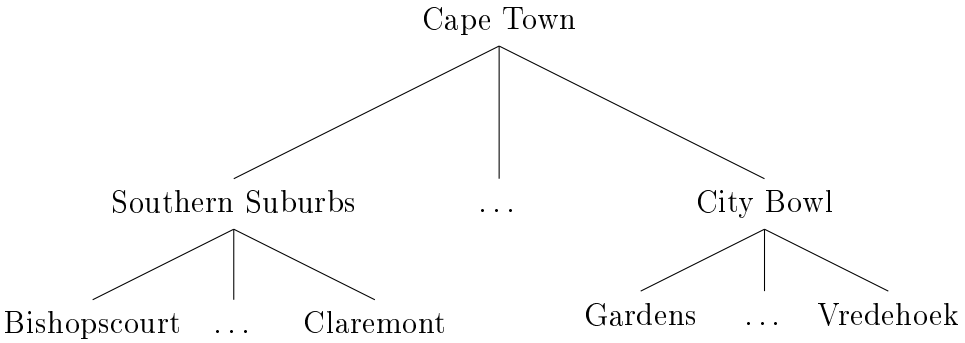
In this section we define how we will structure the generalisation hierarchy for the various attributes making up our crime data. We consider only manual construction of such taxonomies in our implementation. Automated means for generating the attribute taxonomy as given by [42] do exist. Their work introduces the concept of a natural domain generalisation hierarchy (NDGH). However as pointed out in [42] such automated domain generalisation is NP-hard and given the complexity to solve for optimal anonymisation using a genetic algorithm the combination of NDGH with a GA approach would seem less feasible given time constraints in practice. We therefore restrict our focus to constructing manual attribute taxonomies.

The original MCRF by [7] being based on the Datafly algorithm used global recoding after prioritising attributes for generalisation to suit crime data (namely numerical attributes such as age were generalised first) in an attempt to reduce information loss. Our attempt to provide third parties with more information utility is focused on using a local recoding to avoid over-generalisation. We combine this with a weighting scheme on the QIDs as discussed in Section 4.4.3 to provide further flexibility from a third party's perspective. Our manual generalisation hierarchies are however similar to those of the original MCRF and are specified as follows:

Reporter and Age taxonomy respectively:



Suburb taxonomy:



One of the main drawbacks related to manual specification of the attribute taxonomy is knowledge required by a user to select appropriate taxonomies. As the MCRF grows and adapts this problem could be seen to become more accentuated. Ideally we would like to automate the attribute taxonomy especially in a resource constrained environment. The NDGH mentioned above might be an appropriate starting point.

4.4 Information loss

The trade-off between information loss and guaranteeing privacy for crime reporters is one of the central themes of this work. Towards achieving an optimal anonymisation we require metrics for both these inputs. Privacy for our purposes will be defined in terms of the level of k-anonymity and l-diversity in Section 4.6.1 and Section 4.6.2 respectively. The loss metric we define in this section is an adaptation of the original general loss metric (LM) introduced by [24]. Some notation related to defining the general loss metric as given by [24] will be helpful at this stage.

4.4.1 Information loss notation

- A is the attribute space with $a \in A$ where a is one specific attribute (a column in the crime data table)
- $T(a)$ denotes the generalisation taxonomy defined for numerical attributes $a \in A$. Let $T(a)_{max}$ and $T(a)_{min}$ be the upper and lower limits respectively for the generalisation of the numerical attribute $a \in A$. Furthermore define $t_{d,n}(a)_{max}$ and $t_{d,n}(a)_{min}$ as the upper and lower limits of the current generalisation for attribute a in tuple d during iteration n.
- $K(a)$ denotes the generalisation taxonomy defined for categorical attributes $a \in A$. $K(a)_{total}$ is defined as the total number of leave nodes generated by the taxonomy tree $K(a)$. P is the number of nodes created by $K(a)$. $k(a)_p$ is the sub-tree rooted at node $p \in P$. $k(a)_{p,total}$ is the number of leave nodes in the sub-tree rooted at node $p \in P$

4.4.2 The original loss metric

Definition 4.4.1 (Original general information loss)**.** Using our own notation we can now define the general information loss as specified in [24]. Due to both numerical and categorical attributes the loss for attribute a in tuple d is defined piece-wise as follows for iteration n of the GA:

$$IL_{d,n}(a) = \begin{cases} \dfrac{k(a)_{p,total} - 1}{P - 1} & \textit{if categorical} \\ \dfrac{t_{d,n}(a)_{max} - t_{d,n}(a)_{min}}{T(a)_{max} - T(a)_{min}} & \textit{if numerical} \end{cases} \tag{4.1}$$

Definition 4.4.2 (Original loss metric)**.** Our own notation for the loss metric given by [24] yields:

$$LM_n = \sum_{d \in D} \sum_{a \in A} IL_{d,n}(a) \tag{4.2}$$

4.4.3 CrimeGenes loss metric

Here we introduce our weighting scheme for the loss metric which enables end users to prioritise specific attributes during anonymisation. By this we mean that an end user requesting anonymised data can define which of the QIDs should contain more information whilst still adhering to a given privacy requirement. In Section 4.6 we introduce two anonymisation algorithms called "CrimeGenes-kanon" (CG-kanon) and "CrimeGenes-diverse" (CG-kanon) where both implement the same CG loss metric.

Our motivation for weighting the loss metric is that we do not know how the anonymised data will be used. Data mining tools for classification or clustering might be applied to the data. Users may simply want to get more insight into the distribution of a particular attribute. Although the skills shortage in our resource constrained setting lies with the law enforcement agency, third parties performing analytics should not have this problem. They could benefit from being able to modify the anonymisation to best suit their particular use case.

Using notation introduced earlier we define the CG loss metric before returning to issues relating to the practical implementation of introducing such flexibility for third party users: We define the CG loss metric (LM_{CG}) as follows:

Definition 4.4.3 (LM_{CG})**.**

$$LM_{CG,n} = \sum_{d \in D} \sum_{a \in A} w_a \times IL_{d,n}(a) \tag{4.3}$$

Where:

- w_a is the weight assigned to attribute $a \in A$ by the end user. The appropriate weight for an attribute is case specific and usually involves an iterative process. A user's weightings are typically informed by output from previously generated samples (issues around sampling and related disclosure risks are covered later).
- $IL_{d,n}(a)$ as defined earlier

4.5 Crime severity weighting

We introduce a crime weighting scheme to facilitate an automated anonymisation approach based on the severity of an offence. This paper considers guideline sentences for offences as an appropriate weighting mechanism, although some work is still required in a South African context to determine such guidelines.

One such paper by [50] looks at the possibility of guideline sentences for offences in South Africa whilst referencing the efforts of the state of Minnesota in the United States. Minnesota has implemented their guideline sentence framework since 1980 according to [50] and publishes revised guidelines on an annual basis. At the time of writing the latest available guideline titled "Minnesota Sentencing Guidelines & Commentary" was published on 1 August 2013 [1].

[1] Available at: http://www.mn.gov/sentencing-guidelines/images/2013%2520Guidelines.pdf

The complexity of the Minnesota guideline together with different crime categories (compared to South Africa) makes direct application of this guideline impractical for our purposes. However, the framework nevertheless gives valuable insights into how a guideline sentencing framework has been applied in practice. The implementation of such a framework adds weight to our notion of using standard weightings for our anonymisation process.

Crime	Severity
Corruption or Embezzlement	3
Disorderly conduct	3
Vandalism	3
Burglary	5
Drunken Driving	5
Family or Domestic Violence	5
Illegal gambling	5
Theft	5
Other	5
Robbery	7
Assault	7
Drug related	10
Arson	10
Forgery or Fraud	15
Rape	20
Murder	25

Figure 4.3: Crime severity weightings

We denote the crime severity weight as $S(c)$ where $c \in C$. C denotes the list of offences as shown in Figure 4.3 and $S(\cdot)$ maps the crime category to its weight, which is simply the guideline sentence duration (in years) for that crime. For instance $S(theft) = 5$. Since the sentencing duration for a given crime could be interpreted as the severity with which society as a whole views a particular offence we propose using this measure as a proxy for the risk to a crime reporter of privacy loss. For example, the risk of privacy loss for reporting a murder is in general higher than for reporting a misdemeanour.

4.6 Achieving automation

As stipulated in the previous section our automation is premised on the assumption that a crime severity scale exists where such severities are representative for society as a whole. Our implementation is known as "CrimeGenes" (CG) and we explore two automated anonymisaton algorithms for CG which employ a GA approach.

The definition of privacy for the anonymised data is what distinguishes the two algorithms - one algorithm is based on k-anonymity (CG-kanon) and one on l-diversity (CG-diverse). Inferential attacks on anonymised data from CG-kanon required us to consider CG-diverse to enforce stricter privacy constraints.

The crime severity scale facilitates automation but our choice of a GA ensures that information utility of the anonymised data is maximised given the privacy constraints defined by such a crime severity scale. In this section we contrast the crime severity schemes of both CG-kanon and CG-diverse and discuss how the GA component differs between the two approaches.

4.6.1 Understanding CG-kanon

As covered in our review of the literature, the notion of privacy defined by k-anonymity seeks to hide tuples with the same QIDs in a group of k. For CG-kanon we propose using the crime severity scheme in Figure 4.3 to hide tuples for more severe crimes in larger equivalence classes. For instance, we would like a report about a *murder* to satisfy 20-anonymity whereas an incident of *theft* should be shown in an equivalence class satisfying a lower level of say 5-anonymity.

It is important to take note of the following points about CG-kanon:

- Our notion of hiding more severe crimes in larger equivalence classes implies nothing about the *absolute* level of k-anonymity for different crime categories. It is instead a *relative* statement about the level of k-anonymity for different crimes in the anonymised dataset. The severity weighting is converted to a severity penalty which is used by the GA. We elaborate on this later
- An absolute level of minimum k-anonymity is therefore required, we denote this by k_{min}. The purpose of k_{min} is to guarantee a global minimum level of k-anonymity that all equivalence classes must adhere to. The GA then uses k_{min} as a baseline level for k-anonymity whilst moving tuples into appropriate equivalence classes based on their crime severity. k_{min} will be defined in more detail below
- The GA of CG-kanon uses the crime severity scale, k_{min} and the CG loss metric as input parameters for optimisation. The loss metric was introduced in Section 4.4. Once these parameters are provided the GA takes over the entire process until optimisation is complete. Section 4.9 will provide further insights into details of the GA for CG-kanon

4.6.1.1 k_{min} for CG-kanon

We introduced k_{min} above as the minimum level of k-anonymity for the dataset as a whole. Here we define it and comment briefly on its interpretation:

$$k_{min} = max(k_{constant}, min(S_D(\cdot))) \tag{4.4}$$

Where:

- $k_{constant}$ is a fixed minimum level of k, in our implementation this was set at 5
- $S_D(\cdot)$ is the set of all severities for the dataset D

The definition for k_{min} shows that the global minimum level of k-anonymity is set either at a fixed level of $k_{constant}$ or at the lowest level of crime severity in the specific dataset if this is higher than $k_{constant}$. For instance if $k_{constant}$ is set at 5 and the lowest crime severity weight in the dataset is 3 then k_{min} will be 5. However if the lowest crime severity weight in the dataset is 7 then k_{min} will be set to 7. Our definition for k_{min} therefore ensures that an adequate minimum k-anonymity will be set for datasets regardless of the reported crimes in those datasets.

$k_{constant}$ can be seen to be a subjective parameter used to increase the global level of k-anonymity independent of entries in the dataset. By setting $k_{constant}$ too high we will reduce the impact of crime severity on k_{min}. For instance, if $k_{constant} = 10$ datasets with minimum crime severities of 5 or 7 will be anonymised to satisfy 10-anonymity. We therefore suggest keeping $k_{constant}$ close to the lowest crime severity weighting in most cases to allow crime severity to dictate k_{min}. However, if for any reason the need arises to raise the minimum level of k-anonymity, $k_{constant}$ can be set higher for this purpose. For our implementation $k_{constant}$ was set at 5 which was marginally higher than the lowest crime severity of 3. For crime data a minimum 20% chance of positive disclosure seemed more appropriate than a 33% chance using 3-anonymity, however this was purely subjective. It can be seen that how the crime severity scale is defined will have an impact on the appropriate level of $k_{constant}$. Rather than being prescriptive we therefore emphasise that $k_{constant}$ should be set on a case-by-case basis within the appropriate context.

4.6.1.2 The severity penalty

As noted earlier our severity weighting enters the GA of CG-kanon as a severity penalty. This severity penalty function is used by a fitness function during the optimisation process. The fitness function is defined and discussed in Section 4.7.

We call the severity weighting a *severity penalty (SP)* in the context of the fitness function. The GA seeks to minimise the total severity penalty as part of the optimisation process. The severity penalty for a single tuple d is given as follows:

$$SP_d = \frac{S_d(c)}{|e_{d,n}|} \tag{4.5}$$

Where:

- A tuple is written as d and $d \in D(\cdot)$
- $D(\cdot)$ denotes a dataset of crime data. For instance, using notation introduced earlier D(O) is our dataset of reported crime data
- $e \in E$ and E denotes the set of equivalence classes
- $|e_{d,n}|$ is the size of the equivalence class that tuple d finds itself in during iteration n of the GA

We see from Equation 4.5 that severe crimes in small equivalence classes will result in large penalties and vice versa. If for example a report about a murder (with severity weighting of 25) is located in an equivalence class that satisfies 5-anonymity after generalisation, a penalty of $\frac{25}{5} = 5$ is generated. If however an incident about theft is located in that same equivalence class, this will only cause a severity penalty of 1.

The GA for CG-kanon is set up to minimise the total severity penalty across all tuples in the dataset. Therefore if it can find a replacement for the murder incident (say an incident of drunken driving) which reduces the total severity penalty it might do so. For instance if the drunken driving case was located in an equivalence class of size 25 we could move the murder incident to this larger equivalence class if it can share the same QIDs. The drunken driving case could be generalised to the 5-anonymity equivalence class if it can be assigned the appropriate QIDs. Whereas before the combined severity penalty for the two tuples was $\frac{25}{5} + \frac{5}{25} = 5.2$, after swapping the two tuples between equivalence classes the combined penalty is now much lower at $\frac{25}{25} + \frac{5}{5} = 2$.

The above example simplifies the GA approach for CG-kanon considerably since there are other constraints such as information utility which the algorithm evaluates simultaneously. However the example demonstrate how the severity penalty forces the GA to move severe crimes to larger equivalence classes and vice versa.

We mentioned that the total severity penalty for the entire dataset is what the GA seeks to minimise amongst other factors. Equation 4.5 defined the severity penalty for a single tuple. The aggregate severity penalty for a dataset which the fitness function targets is denoted by $SP_{tot,n}$ and calculated as:

$$SP_{tot,n} = \sum_{d \in D} SP_{d,n} \tag{4.6}$$

The equation shows that the total severity penalty for a CG-kanon dataset is simply the summation of the severity penalties of the individual tuples.

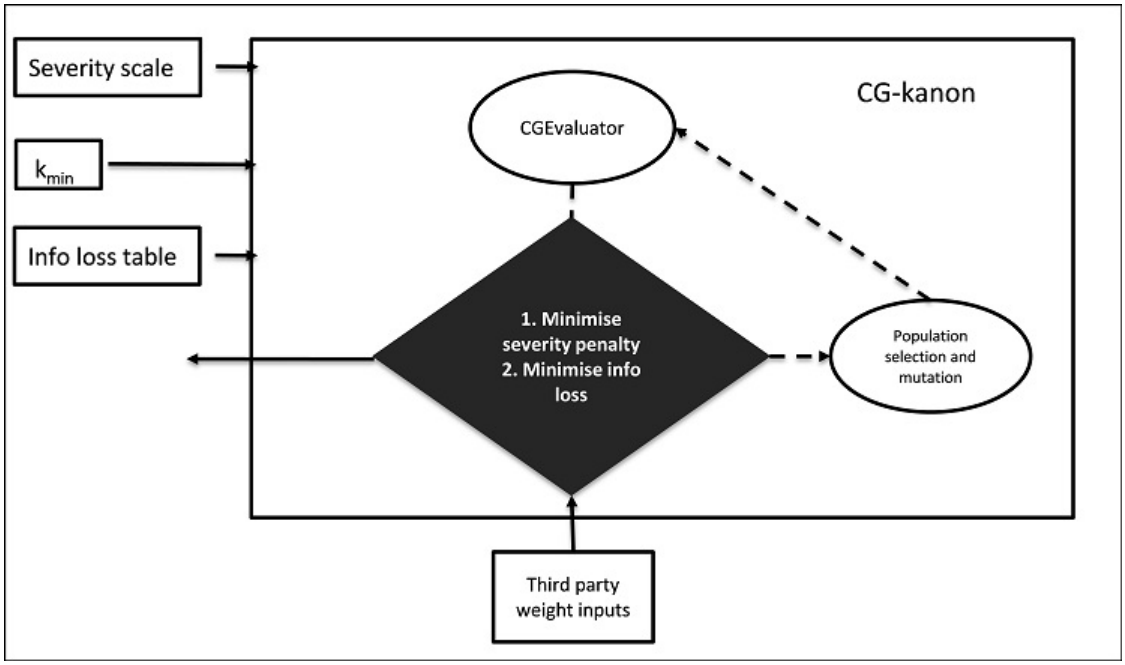


Figure 4.4: CG-kanon

Figure 4.4 provides a graphical representation of the CG-kanon GA. It is based on a selection of components from Figure 3.2 and shows how the parameters for CG-kanon fit into the broader algorithm. The information loss table shown in the diagram is a lookup table for losses generated using LM_{CG}. We provide a UML diagram of the CG-kanon GA for the interested reader in Appendix A.

4.6.1.3 Limitations of CG-kanon

The main drawback of CG-kanon is that diversity of the sensitive attribute in large equivalence classes may be quite low depending on the distribution of tuples in the dataset. This becomes apparent in Section 5.2.1.1 of our results.

The algorithm might also suppress a large proportion of tuples in an attempt to satisfy the minimum k_{min}-anonymity for the entire dataset. To counter this within CG-kanon we limit the size of the largest equivalence class by introducing an additional penalty into the fitness function if an equivalence class is larger than a specified limit. This forces the GA in CG-kanon to reduce the size of the largest equivalence classes. Reducing the size of the largest equivalence classes is a pragmatic approach to help improve diversity of the sensitive attribute for those larger equivalence classes. We also found that this decreased the number of suppressions created by the GA to satisfy the minimum k_{min}-anonymity.

Nevertheless, limiting the size of the largest equivalence classes is an artificial approach to compensate for the well-know vulnerability of k-anonymity to inference attacks. This vulnerability was accentuated by using the severity penalty as part of CG-kanon on our generated data. Our CG-diverse approach therefore seeks to address this limitation by seeking to achieve l-diversity directly rather than applying it as an extension to k-anonymity. The average severity metric provided a means to automate the anonymisation process within the context of l-diversity.

4.6.2 Understanding CG-diverse

Throughout most of the literature l-diversity is considered as a further privacy extension to k-anonymity. In this regard algorithms are often adapted from meeting k-anonymity to also meet the stricter l-diversity definition. In a GA context, however, this approach would make the fitness function increasingly complex and as Section 5.2.3.3 of our results show this causes inefficiencies. Our approach for CG-diverse was therefore to use the notion of diversity as our starting point and from this obtain a minimum level of k-anonymity which is also adhered to. This is because an l-diverse dataset is also at least k-anonymous such that k is equal to l.

Whereas CG-kanon used the crime severity scale as a severity penalty for the GA , CG-diverse uses the crime severity scale to calculate an average crime severity for the dataset and individual equivalence classes during a two-step process.

4.6.2.1 Average severity

We distinguish between two average severity (AS) measures. One is calculated based on the entire dataset and the other on a specific equivalence class. The next section will show how the two definitions are used by CG-diverse.

The AS for the entire dataset is defined as follows, using notation introduced earlier:

$$AS_D = \frac{\sum_{d \in D} S_d(c)}{|D|} \tag{4.7}$$

In a similar manner the AS for a specific equivalence class is calculated as:

$$AS_e = \frac{\sum_{d \in e} S_d(c)}{|e|} \tag{4.8}$$

Figure 4.5 shows these average severity measures as calculated for a given sample dataset. For instance the AS_e for equivalence class 2 is 5 ($\frac{5+3+7+7+7}{5}$). We also see from looking at the crime

severity scale in Figure 4.3 that the AS for both Equation 4.7 and Equation 4.8 can range between 3 and 25. We now look to see what this implies for our CG-diverse algorithm.

4.6.2.2 Diversity in two steps

The NP-hardness of achieving l-diversity for a given dataset requires us to consider a trade-off between complete optimality for an l-diverse dataset and achieving results in a reasonable time period. Implementing CG-diverse as a two-step process proved to provide satisfactory results and addressed the vulnerability to inferential attacks of CG-kanon as our results show in Chapter 5. The two steps are:

1. The AS_D is calculated for the entire dataset. This AS_D is used by the GA of CG-diverse as an input parameter (again the loss metric from Section 4.4.3 is also an input as for CG-kanon). The GA starts its optimisation and anonymises the dataset with a target level of l-diversity such that l is equal to AS_D. Therefore if AS_D is higher (i.e. more severe crimes are present in the dataset) the level of l-diversity targeted for the data as a whole will also be higher. As noted earlier, however, given the severity weighting scale in Figure 4.3 the level of l-diversity in our case will be restricted to the range between 3 and 25 depending on the underlying dataset.

   The stochastic nature of a GA means that there is no guarantee that the level of AS_D-diversity will be achieved during the allocated runtime. Therefore since we operate within a time constrained environment we will need to interrupt the optimisation process after a specified amount of time as Section 4.7.1 will discuss.

   We could at this point suppress all equivalence classes that do not meet the required level of AS_D-diversity. However this would result in significant information loss. There may be a large number of equivalence classes where the constituent tuples contain only less severe crimes. Requiring those equivalence classes to meet the same global level of AS_D-diversity may be overly restrictive. We can see this by considering some sample results in Figure 4.5. In this specific case we had AS_D equal to 11 for the entire dataset. Equivalence class (EC) 4 satisfies the requirements set by our first step (i.e. diversity is greater than or equal to AS_D). However all other ECs would have to be suppressed on this basis.

   Looking at EC 1 we see that it is 6-diverse and 7-anonymous. The crimes in EC 1 are also in general not too severe with *Robbery* as the most severe crime with severity 7. Step 2 of CG-diverse is designed to identify such ECs with lower average severity (but adequate relative diversity) and avoid them being suppressed. This is done by using AS_e.

Age	Suburb	Reporter	Crime	Diversity (equivalence class)	AS (dataset)	AS (equivalence class)
Equivalence class 1						
18_22	Cape Town	Proxy	Drunken Driving	6	11	5.5
18_22	Cape Town	Proxy	Drug related	6	11	5.5
18_22	Cape Town	Proxy	Disorderly conduct	6	11	5.5
18_22	Cape Town	Proxy	Theft	6	11	5.5
18_22	Cape Town	Proxy	Corruption or Embezzlement	6	11	5.5
18_22	Cape Town	Proxy	Robbery	6	11	5.5
18_22	Cape Town	Proxy	Robbery	6	11	5.5
Equivalence class 2						
18_22	Cape Town	Reporter	Drunken Driving	4	11	5.0
18_22	Cape Town	Reporter	Corruption or Embezzlement	4	11	5.0
18_22	Cape Town	Reporter	Robbery	4	11	5.0
18_22	Cape Town	Reporter	Theft	4	11	5.0
18_22	Cape Town	Reporter	Theft	4	11	5.0
Equivalence class 3						
18_22	Cape Town	Witness	Other	8	11	7.5
18_22	Cape Town	Witness	Family or Domestic Violence	8	11	7.5
18_22	Cape Town	Witness	Other	8	11	7.5
18_22	Cape Town	Witness	Theft	8	11	7.5
18_22	Cape Town	Witness	Rape	8	11	7.5
18_22	Cape Town	Witness	Burglary	8	11	7.5
18_22	Cape Town	Witness	Drug related	8	11	7.5
18_22	Cape Town	Witness	Assault	8	11	7.5
18_22	Cape Town	Witness	Vandalism	8	11	7.5
Equivalence class 4						
18_87	City Bowl	Proxy	Theft	11	11	8.5
18_87	City Bowl	Proxy	Other	11	11	8.5
18_87	City Bowl	Proxy	Murder	11	11	8.5
18_87	City Bowl	Proxy	Arson	11	11	8.5
18_87	City Bowl	Proxy	Drunken Driving	11	11	8.5
18_87	City Bowl	Proxy	Other	11	11	8.5
18_87	City Bowl	Proxy	Forgery or Fraud	11	11	8.5
18_87	City Bowl	Proxy	Family or Domestic Violence	11	11	8.5
18_87	City Bowl	Proxy	Drug related	11	11	8.5
18_87	City Bowl	Proxy	Burglary	11	11	8.5
18_87	City Bowl	Proxy	Drug related	11	11	8.5
18_87	City Bowl	Proxy	Illegal gambling	11	11	8.5
18_87	City Bowl	Proxy	Drunken Driving	11	11	8.5
18_87	City Bowl	Proxy	Disorderly conduct	11	11	8.5
Equivalence class 5						
58_62	Atlantic Seaboard	Reporter	Disorderly conduct	6	11	5.5
58_62	Atlantic Seaboard	Reporter	Theft	6	11	5.5
58_62	Atlantic Seaboard	Reporter	Drunken Driving	6	11	5.5
58_62	Atlantic Seaboard	Reporter	Drug related	6	11	5.5
58_62	Atlantic Seaboard	Reporter	Corruption or Embezzlement	6	11	5.5
58_62	Atlantic Seaboard	Reporter	Robbery	6	11	5.5
Equivalence class 6						
58_62	Cape Town	Reporter	Vandalism	5	11	5.0
58_62	Cape Town	Reporter	Drunken Driving	5	11	5.0
58_62	Cape Town	Reporter	Illegal gambling	5	11	5.0
58_62	Cape Town	Reporter	Burglary	5	11	5.0
58_62	Cape Town	Reporter	Assault	5	11	5.0

Figure 4.5: Average severity versus diversity

2. During step 1 our GA for CG-diverse generated an anonymised dataset optimised for data utility whilst constructing equivalence classes that meet AS_D-diversity. The anonymised dataset after step 1 provides us with an optimal result given metrics for information loss and the time we allocated for it to complete.

   Step 2 seeks to exploit the optimisation further by assessing the privacy of individual equivalences classes not meeting the global AS_D-diversity requirement. Here AS_e from Equation 4.8 is used to compare the level of diversity for an equivalence class to its specific average severity. If the average severity of an equivalence class exceeds the level of diversity in that equivalence class, these tuples are generalised to the highest level. EC 2 in Figure 4.5 is an example of where this would occur. Alternatively tuples where diversity exceeds AS_e are published as is (EC 1, 3, 5 and 6). EC 4 was already validated during step 1 as it satisfied AS_D-diversity.

   Note that during the first step we used a global severity measure for all equivalence classes but in this step we reassess whether equivalences classes not meeting the global severity of AS_D should in fact be suppressed based on their local individual characteristics as defined by AS_e.

   Lastly, we note that this second step is computationally inexpensive as it simply requires us to compare AS_e with the actual observed diversity of the equivalence class. Since both these quantities are calculated during the optimisation in step 1, processing for step 2 takes a few milliseconds.

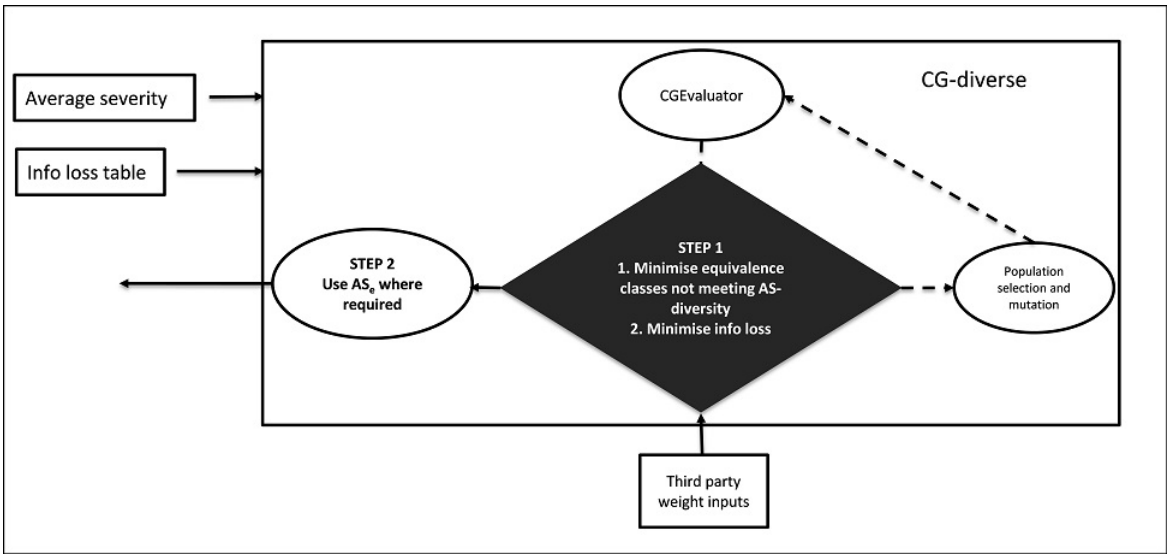


Figure 4.6: CG-diverse

We see from Figure 4.6 that CG-diverse uses the same CG loss metric as CG-kanon. Also whilst CG-kanon has three main inputs (k_{min},SP_D and LM_{CG}) for the GA, CG-diverse only

has two inputs, namely AS_D and LM_{CG}. AS_e is not an input to the GA itself since it is used post optimisation. The fact that CG-diverse has fewer inputs and therefore a simpler fitness function increases optimisation efficiency as we will show in our results.

4.7 Fitness function

The fitness function is used during each iteration to evaluate how well the proposed solution meets the desired outcomes. The "fitness" of each individual is measured against an objective or a set of objectives as in our case. The fitness function in our case evaluates multiple objectives such that for CG-kanon:

$$FF_n^{CG-kanon} \propto \frac{1}{f(SP_{tot,n}, LM_{CG,n})} \tag{4.9}$$

and for CG-diverse:

$$FF_n^{CG-diverse} \propto \frac{1}{f(AS_{D,n}, LM_{CG,n})} \tag{4.10}$$

The above definitions show that the fitness of an individual is inversely related to the information loss and the respective severity measures for CG-kanon and CG-diverse. Our fitness function is located in the CGEvaluator as shown earlier in Figure 3.2. Specifying the fitness function may be an iterative process and in Section 5.1 we discuss some of the challenges experienced in this regard. Whilst $SP_{tot,n}$,$AS_{D,n}$ and $LM_{CG,n}$ are the main drivers of the fitness functions we should point out that within our constrained environment additional parameters were introduced as covered in Section 5.1 of our results to make the implementation practical.

4.7.1 Stopping criteria for CG

Our literature review covered stopping criteria relevant to our implementation. The first section of Chapter 5 of our results covers pilot studies that we used to understand how best to implement our GA for CG given our resource constrained environment.

It became evident from observing the convergence of parameters in the fitness function that the appropriate factor for terminating the optimisation is time (i.e. runtime of the algorithm). By using a Pareto optimal GA approach we are unable to provide any guarantees about when the specified level of k_{min} and AS_D would be satisfied (for CG-kanon and CG-diverse respectively) by all equivalence classes. Third parties using the data also cannot wait for extended time periods to extract data.

We therefore adopted a pragmatic approach based on our experimental results. We introduced

a sampling scheme to reduce the computational complexity of each run. By looking at the optimisation efficiency of CG-kanon and CG-diverse we observed that a sample size of 1000 crime reports provided useful results in 30 minutes. A more detailed discussion of our sampling scheme follows in the next section.

4.7.2 CG sampling scheme

Our motivation for implementing a sampling scheme is two-fold, namely:

1. To reduce computational complexity of the GA and thereby achieve optimal results within a shorter time-frame
2. The attribute weighting scheme introduced for the CG loss metric introduces additional disclosure risks which could be addressed by an appropriate sampling scheme

The rationale behind our first point is self explanatory. We therefore discuss the second point in more detail.

4.7.2.1 Implications of introducing flexibility through LM_{CG}

A loss metric which gives third parties flexibility to weight the QIDs brings with it some complications. Most notably it increases disclosure risk as a user might iteratively overweight and underweight QIDs to obtain different sets of anonymised data. These datasets could then be combined to achieve a disclosure. Consider the scenario where a third party analyst draws three anonymised samples of 1000 records. For each sample 1, 2 and 3 he recursively overweights *Age*, *Suburb* and *ReporterType* respectively. For sample 1, 2 and 3 this gives much more granular information for *Age*, *Suburb* and *Reportertype* respectively. If sample sizes of 1000 records are drawn randomly the probability of disclosure increases significantly if he repeats the above process. This increased disclosure risk holds even where an attribute weighting scheme is not introduced since the anonymisation for each record is not unique within different samples.

One approach to overcome this problem is to use sampling without replacement. This may seem overly restrictive, however, as the database grows and larger samples can be drawn a more representative view of the data is obtainable through a sampling approach if this is done randomly. Whilst a sampling approach addresses the disclosure concern raised above, it also solves two further problems related to our GA implementation:

- **Computational complexity**. A smaller dataset makes a GA much faster and more feasible for practical consideration in our resource constrained environment

- **New crime reports**. New data can be combined with existing data from where samples are being drawn. This makes most recent crime reports available for selection and facilitates growth in the volume of anonymised data available to third parties. As covered in Chapter 2 sequential releases bring with it some complications. However sampling without replacement circumvents most of these issues albeit in a rather crude manner

There are some drawbacks of the above sampling scheme and these include:

- Third parties are restricted in the number of samples they can draw
- In cases where a third party has depleted the number of samples he can draw, new crime reports are not immediately available to be released as part of the dataset. The third party will be required to wait until adequate new reports have entered the database to create a new sample for release
- The impact of the point above is that time selection is introduced where later samples are drawn by a user
- The system needs to keep track of records already released to a third party and remove these from those records available to that party in subsequent draws
- The risk of disclosure from collaboration between third parties is increased. However this should be controlled by confidentiality agreements and terms-of-use contracts as a default precaution

Our results in Chapter 5 give further insights into the attribute weighting scheme and the above sampling scheme.

4.8 Countering biases

Before proceeding with this section it is important to point out that whilst our work identifies the notion of biases in the MCRF we were unable to fully consider the issue during our implementation. We do provide possible approaches of dealing with it by incorporating an adjustment factor. We also conducted preliminary experiments in isolation to the rest of the CG implementation to gauge the validity of our proposed adjustment factor and we comment on these experiments in Chapter 5. However the scope of our work pertaining to the adjustment factor for biases are not as extensive as our focus on CG-kanon and CG-diverse for instance. This could therefore be an interesting area of future research.

4.8.1 The adjustment factor

The notion of an adjustment factor aims to compensate for possible biases introduced through the MCRF. We previously defined a bias as an instance where the reported frequency distribution of crime reports differs from the true frequency distribution. Section 4.1.2 pointed out various biases (distortions) that may be introduced into the crime data through a MCRF - these biases are what we seek to minimise through an adjustment factor.

The validity of our idea is firstly premised on the notion that we are able to determine the true distribution of crimes. Secondly we are assuming that third parties are interested in the true underlying crime statistics and not only in crime data gathered from the MCRF.

In practice we may take the true underlying distribution of offences to be historical data gathered by the law enforcement agency over several years. Admittedly, in the context of a developing country it may be less likely that such data is accurate or even exists. Furthermore, the true distribution needs to be reflective of the area covered by the MCRF. Nevertheless for our purposes we suppose that we can specify a true distribution to serve as a benchmark for assessing biases in reported data. We might also consider adapting our true distribution over time to incorporate the reported data and adjust our conditional probabilities in a Bayesian manner.

Whilst our crime severity scale was used in CG-kanon and CG-diverse to increase privacy for more serious crimes, the adjustment factor favours generalisation of tuples that are likely to be biased. Intuitively less information is lost with respect to the true distribution of offences when distorted data are generalised.

We turn to methods used for statistical hypothesis testing to assess whether the reported frequency of offences deviate from those of the true distribution. The χ^2 goodness of fit test serves as a possible candidate to formalise this notion.

4.8.2 The χ^2 test

We introduce the following notation related to the number of actual crime reports and those derived from the true population:

- T_c is defined as the true number of offences in category c where $c = 1, 2...n \in N$
- T represents the total true number of offences across all categories

Notation for frequency counts can now be stated using the notation above:

- $f_c^{obs} = \frac{O_c}{O}$ is the reported frequency count for offence in category c
- $f_c^{true} = \frac{T_c}{T}$ is the true frequency count for offence in category c
- E_c is the expected number of reported offences for category c based on the true distribution. This is defined as $E_c = f_c^{true} \times O$

Definition 4.8.1 (χ^2 distribution)**.** The summation of Z_c^2 where $c = 1, 2...c \in C$ have a χ^2 distribution with n degrees of freedom if and only if $Z_c, c = 1, 2...c \in C$ are identically and independently distributed as standard normal random variables. Formally,

$$\sum_{c \in C}^{n} Z_c^2 \backsim \chi_n^2, \ \textit{where } Z_c \backsim N(0,1) \tag{4.11}$$

The result above is most often applied to assess goodness-of-fit across a number of categories C simultaneously. Using our notation above we could test whether a significant difference (in a statistical sense) exists between the true and reported frequencies of crime reports. Such a goodness-of-fit test is defined next using our notation above.

Definition 4.8.2 (χ^2 goodness-of-fit test)**.** From the statistical literature on hypothesis testing we know that the random variable Z_c as defined below has a $N(0,1)$ distribution:

$$z_c = \frac{(O_c - E_c)^2}{E_c} \tag{4.12}$$

The summation across all crime categories then gives us a χ^2 random variable according to Equation 4.11 to be used for goodness-of-fit, such that the test statistic is written as:

$$\sum_{c \in C} z_c \backsim \chi_{|C|}^2 \tag{4.13}$$

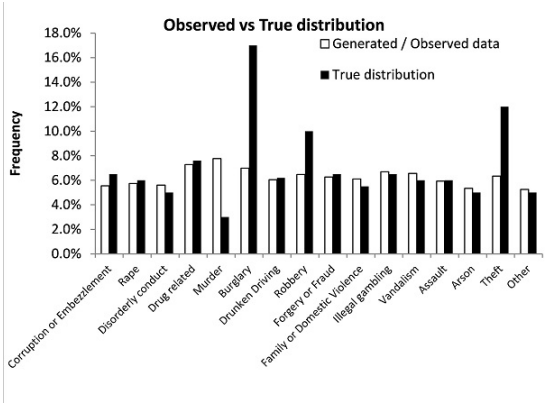
However, more than being able to determine whether the observed and true distributions are alike, we want a metric to incorporate into the fitness function of a GA which will compensate for specific biases. The test statistic in its current from Equation 4.13 is not suited for this purpose as it can only tell us whether two distributions are similar. Therefore whilst the χ^2 test is widely accepted as a sound metric for determining goodness-of-fit it does not provide the desired result for our GA in its current form. We may however think to adapt it slightly. This is done in the following section where we propose a definition for the adjustment factor.

4.8.3 Defining the adjustment factor

We stated above that within the context of a GA the test statistic from a χ^2 test is not suitable. However if we consider the contribution of each individual crime category to the overall test statistic, namely z_c, as our adjustment factor we may have a suitable answer. We consider an example to demonstrate this.

Figure 4.7(a) compares the distribution of crime reports used for CG (observed) with an assumed true distribution of crimes. The table in Figure 4.8.3 shows how we calculated the adjustment factor for each crime category as $adj_c = \sqrt{z_c}$. These results show that adj_c was higher where differences between the observed and expected frequencies were most pronounced, namely *Murder*, *Burglary*, *Robbery* and *Theft*. This result can also be confirmed by considering the numerator of Equation 4.12.

During earlier sections of our implementation we showed how different parameters enter the fitness function of the GA as penalty functions. We can now see how the adj_c factors can similarly enter the fitness function to be incorporated into the optimisation process.



(a) True distribution

Crime	Generated / Observed data	True distribution	Z_c	adj_c
Corruption or Embezzlement	555	650	13.88	4
Rape	574	600	1.13	1
Disorderly conduct	561	500	7.44	3
Drug related	729	760	1.26	1
Murder	777	300	758.43	28
Burglary	699	1700	589.41	24
Drunken Driving	605	620	0.36	1
Robbery	648	1000	123.90	11
Forgery or Fraud	627	650	0.81	1
Family or Domestic Violence	611	550	6.77	3
Illegal gambling	670	650	0.62	1
Vandalism	656	600	5.23	2
Assault	593	600	0.08	0
Arson	535	500	2.45	2
Theft	634	1200	266.96	16
Other	526	500	1.35	1

(b) Deriving adjustment factors

Figure 4.7: Adjustment factors

Whilst the derivation for the adjustment factor might seem unnecessarily complicated, we prefer proposing a metric with some statistical grounding for theoretical soundness. Further research might suggest that much simpler heuristic measures to counter biases work equally well. Nevertheless our approach provides a starting point for such discussions.

Whilst we introduce the notion of an adjustment factor in some detail we reiterate that our experimental results are comparatively more limited. We will briefly discuss our findings from such experiments in Chapter 5.

4.9 Anonymisation algorithms

Figure 3.2 gave a broad overview of the anonymisation engine. Now equipped with a well-defined fitness function we set out the algorithms used for optimisation. Some additional terminology specific to GAs might be useful at this stage before looking at the algorithms in detail.

Some GA terminology:

- **Gene.** In our context a gene is equivalent to the QID and the corresponding crime for a single tuple
- **Genotype.** Traditionally a chromosome defines an individual in the context of GAs. The Opt4J framework refers to the chromosome as the genotype. A genotype in CrimeGenes will be the collection of all genes (i.e. all tuples consisting of a QID and the corresponding sensitive attribute)
- **Adults.** These are individuals which are allowed to reproduce offspring. In our work each individual consists of an entirely generalised data sample which has been anonymised
- **Offspring.** Adults produce offspring using cross-over operations defined below. Offspring constitute data generalisations in subsequent iterations of the algorithm which have been derived by keeping desirable properties of their parent data generalisations
- **Tournament.** A number of multi-objective algorithms use tournaments to improve selection of dominant individuals. The SPEA2 algorithm in our implementation also uses tournament selection. Here a number of individuals are selected to compete for superiority. They are ranked according to fitness and only a selection of the fittest individuals are allowed to generate offspring
- **Cross-over.** This is a GA operator which defines how much genetic material from each adult is taken to produce offspring. The cross-over operator is often distinguished from

mutation which is an operator that introduces random variation into the genetic material

- **Population.** The maximum number of individuals (adults plus offspring) allowed during an iteration.

We comment on how these concepts are implemented in the CrimeGenes GA when we specify the algorithms below.

4.9.1 Anonymisation algorithms

In this section we break down the GAs of CG-kanon and CG-diverse into four main algorithms. Before looking at the finer details it may be helpful to consider the context of each algorithm within the broader framework of our CG implementation. Figure 4.8 graphically shows how each algorithm discussed in this section relates to one another and the broader framework as well. A similar diagram was given in Chapter 3 where the design of CG was given.

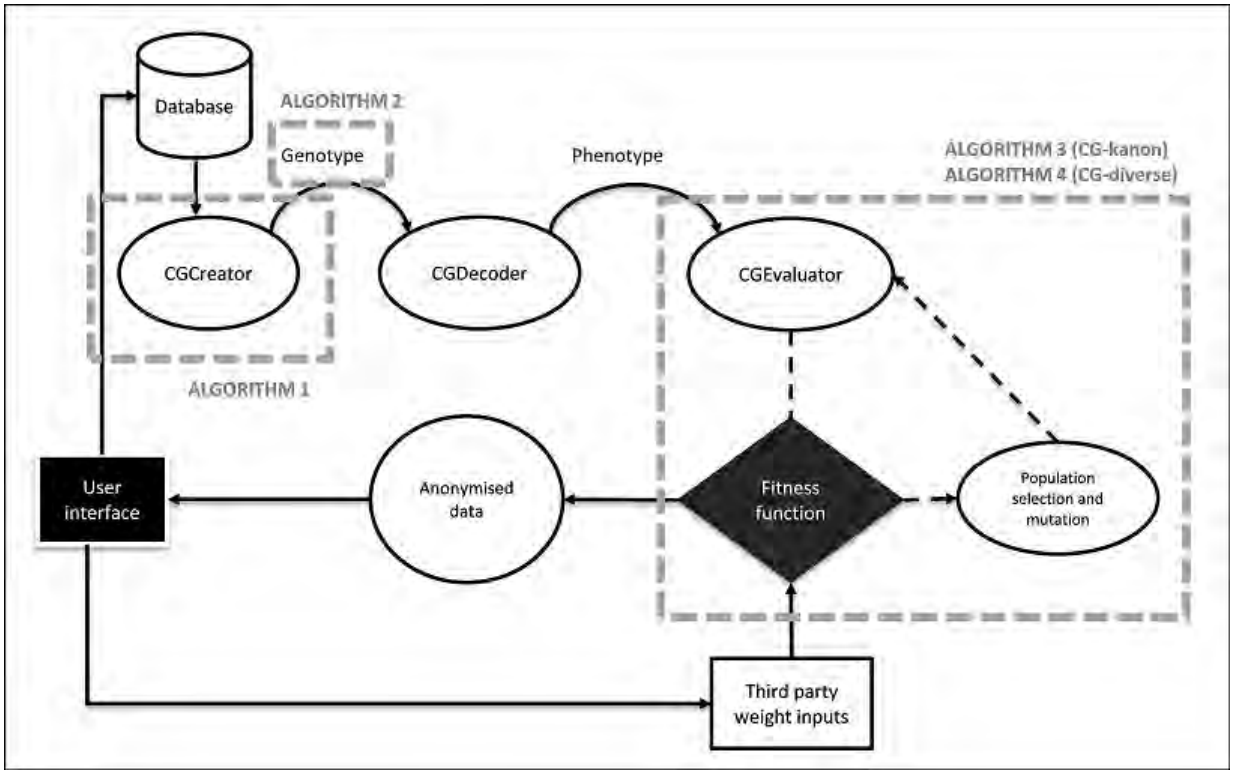


Figure 4.8: Relationship between algorithms

We now look at specifics of each of the four algorithms.

4.9.1.1 Algorithm 1 (Gene creator)

Algorithm 1 is called once at the start of each optimisation run. It generates the gene pool available for creation of the genotype. In our setting a gene is created by combining one permitted generalisation for a tuple with the reported crime. For example, if an incident involving theft was reported by a victim aged 20, living in Newlands an acceptable gene for this report would be "20#Southern Suburbs#Reporter#Theft". There are clearly a number of combinations across the three attribute taxonomies. The purpose of the GeneCreator is to search for all these possible permutations and make them available for use in the GenotypeCreator in Algorithm 2.

Algorithm 1 requires the following inputs: CrimeData, ageHierarchy, addressHierarchy, reporterHierarchy. Each of these are generated from the database using SQL queries and provided as inputs to the algorithm.

Algorithm 1 GeneCreator

Purpose: Creates the gene pool of possible tuple generalisations

Requires: CrimeData;ageHierarchy;addressHierarchy;reporterHierarchy

```
1: crimeKeys ← HashMap < empty >
2: crimeCount ← HashMap < empty >
3: availableGeneralisations ← HashMap < empty >
4: for report in CrimeData do
5:     reportKey ← report(age)#report(address)#report(reporter)#report(crime)
6:     if reportKey not in crimeKeys.keys() then
7:         Add reportKey to crimeKeys
8:         Add report(id) to crimeKeys[reportKey]
9:         crimeCount ← 1
10:    else
11:        Add report(id) to crimeKeys[reportKey]
12:        crimeCount ← crimeCount + 1
13:    end if
14:    availableAges ← ageHierarchy(reportKey)
15:    availableAddress ← ageHierarchy(reportKey)
16:    availableReporter ← ageHierarchy(reportKey)
17:    possibleGeneralisations ← ArrayList
18:    for age in availableAges do
19:        for address in availableAddress do
20:            for reporter in availableReporter do
21:                tempGeneralisation ← age#address#reporter#report(crime)
22:                Add tempGeneralisation to possibleGeneralisations
23:            end for
24:        end for
25:    end for
26:    Add possibleGeneralisations to availableGeneralisations
27: end for
```

4.9.1.2 Algorithm 2 (Genotype)

Algorithm 2 GenotypeCreator

Purpose: Create the genotype for an individual

Requires: GeneCreator,SelectMapGenotype,CompositeGenotype

1: *genePool* ← *ArrayList* ▷ of type SelectMapGenotype
2: *mappedGenePool* ← *ArrayList* ▷ of type SelectMapGenotype
3: *genotype* ← *AnonGenotype* ▷ implements CompositeGenotype
4: **for** *key* in *crimeKeys* **do**
5: Add *crimeKeys*(*key*),*availableGeneralisations*(*key*) to *genePool*
6: **end for**
7: **for** *entry* in *genePool* **do**
8: *entry.init*(*random*) ▷ init() randomly maps report ID to equivalence class
9: Add *entry* to *mappedGenePool*
10: **end for**
11: *genotype* ← *mappedGenePool* ▷ Creates a Genotype of type CompositeGenotype

Algorithm 2 constructs one genotype from the gene pool created in Algorithm 1. It is required that each genotype is created randomly to induce sufficient randomness in the starting population for the entire search space to be adequately explored by the GA. In our implementation the genotype is equivalent to an individual and the genotype can enter the evaluation phase as is. The Opt4J framework in [37] does, however, provide an intermediate decoding module that transforms a genotype into something interpretable in the evaluation module. An example where this may be required is where the genes are binary representations of an object. The decoder module would then transform such binary representations into an object whose features can be evaluated against the fitness function. We did explore the possibility of representing the genes in Algorithm 1 as binary strings in an attempt to reduce memory overhead, however we did not find a noticeable reduction in computation efficiency. In our implementation therefore the genotype is already in a usable form for the evaluation module and the decoder simply creates a copy of the genotype.

4.9.1.3 Algorithm 3 and 4 (CG-kanon and CG-diverse)

With Algorithm 1 and Algorithm 2 defined we are now in a position to set out the full GAs for CG-kanon in Algorithm 3 and CG-diverse in Algorithm 4 responsible for the anonymisation process. The required *userInput* in Algorithm 3 and Algorithm 4 refers to the anonymisation

weighting for attributes provided by the third party through the user interface when requesting anonymised data.

Algorithm 3 CG-kanon

Purpose: Create the CG-kanon genotype for an individual

Requires: GeneCreator,GenotypeCreator, userInput

```
1: anonymisedData ← ArrayList
2: runTime ← t                          ▷ optimisation runtime set in seconds
3: populationSize ← size                ▷ population size for GA is pre-defined
4: GeneCreator
5: for i ≤ populationSize do
6:     GenotypeCreator
7: end for
8: while time < runTime do              ▷ starts the evaluator
9:     function FITNESS FUNCTION(Genotypes)
10:        Evaluate: loss metric
11:        Evaluate: severity penalty
12:        Evaluate: k-value
13:    end function
14:    function CROSSOVER(Genotypes, Fitness)
15:        Crossover
16:        Run tournament               ▷ Apply SPEA2 selector
17:    end function
18:    Update population
19: end while
20: Email anonymised data to user
```

Algorithm 4 CG-diverse

Purpose: Create the CG-diverse genotype for an individual

Requires: GeneCreator,GenotypeCreator, userInput

```
1: anonymisedData ← ArrayList
2: ecDiversity ← HashMap
3: runTime ← t                                  ▷ optimisation runtime set in seconds
4: populationSize ← size                        ▷ population size for GA is pre-defined
5: GeneCreator
6: for i ≤ populationSize do
7:     GenotypeCreator
8: end for
9: calculate AS_D                               ▷ Average severity for data to be anonymised
10: while time < runTime do                     ▷ STEP1:starts the evaluator with global average severity
11:     function FITNESS FUNCTION(Genotypes)
12:         Evaluate: loss metric
13:         Evaluate: diversity
14:     end function
15:     function CROSSOVER(Genotypes, Fitness)
16:         Crossover
17:         Run tournament                      ▷ Apply SPEA2 selector
18:     end function
19:     Update population
20:     Update: ecDiversity                     ▷ Record the diversity of each equivalence class
21: end while
22: for e ∈ E do                                ▷ STEP2: Iterate through equivalence classes for local severity
23:     calculate AS_EC_e
24:     if AS_EC_e ≤ ecDiversity(e) then
25:         Generalise QIDs to highest level
26:     end if
27: end for
28: Email anonymised data to user
```

4.10 Evaluation of CrimeGenes

Our evaluation of the CrimeGenes implementation is informed by our discussion in Section 2.4 of the literature review. In particular rather than using one specific measure to monitor anonymi-

sation we look at the results from different perspectives. This is in line with the notion that third parties will also not restrict their analysis to one particular metric. Our evaluation can be grouped under the following headings:

4.10.1 Qualitative assessment

We assess the quality of an anonymisation from a privacy perspective by looking at the distribution of the sensitive attribute after applying the CG-kanon and CG-diverse algorithms. We also consider the granularity of generalised QIDs to determine whether the attribute weighting scheme performs as expected. Our approach here is similar to that of the Cornell Anonymisation Tool-kit[2] whereby we prefer inspection of the statistical properties of the anonymisation. This often provides valuable insights into the privacy risks related to the anonymised data which may be overlooked by only looking at a summary statistic.

4.10.2 Quantitative

Our quantitative analysis of the results is more in line with mainstream literature. We look at the information loss produced by both CG-kanon and CG-diverse based on the CrimeGenes loss metric (LM_{CG}). Our analysis of the loss metric is also utilised to compare the optimisation efficiency of CG-kanon relative to CG-diverse.

Classification accuracy is measured on a variety of datasets after applying different anonymisation algorithms. The UTD[3] anonymisation toolbox was used to generate anonymised data from the Datafly and Mondrian algorithms for comparison with CrimeGenes. The Weka software framework was then employed to perform classification tests on all anonymised data using a Naive Bayes classifier. A 10-fold cross-validation test was conducted for classification accuracy.

Since our focus for CrimeGenes was not aimed specifically at generating anonymised data for classification we also analysed our results using the Kullman-Leibler divergence. A script was written in Python to test anonymised datasets using this metric as this provides a standardised measure of how different the distribution of the anonymised data are from the distribution of the original data. The KL-divergence assumes that the crime reports follow some multivariate distribution. The original data is taken to be this distribution. The anonymised data is then seen as an empirical distribution taken from that distribution. The KL-value measures the "closeness" of this empirical distribution to the original distribution. However since our generalisations constitute a range of values (e.g. 18_23) for the QID attributes rather than

[2]Available at: http://anony-tool-kit.sourceforge.net/

[3]Available at:http://cs.utdallas.edu/dspl/cgi-bin/toolbox/index.php

point values (e.g. 18) we need to make an assumption about the distribution of point values within those ranges to apply the KL-measure. We assumed a uniform distribution for such point values and applied the following formula to calculate the KL-value:

$$KL = \sum_{n \in N} f(\overrightarrow{x_n}) log \frac{f(\overrightarrow{x_n})}{f'(\overrightarrow{x_n})} \tag{4.14}$$

where:

- $\overrightarrow{x_n}$ is the vector of QIDs for tuple n
- $f'(\cdot)$ is the empirical distribution obtained from the anonymised data, such that $f'(\overrightarrow{x_n}) = \frac{1}{N} \times |e| \times Pr[\overrightarrow{x_n}|E = e]$
- E is the set of equivalence classes

4.11 Discussion

It is evident from our implementation that the crime severity scale is the crucial component to ensuring automation for both CG-kanon and CG-diverse. The GAs in CG-kanon and CG-diverse implement two different input parameters ($SP.$ and $AS.$ respectively) derived from this crime severity scale.

It is important to reiterate that the GAs of CG-kanon and CG-diverse handle all aspects of the anonymisation and optimisation process once it receives the required input parameters and starts running. This is because the framework of a GA enables us to specify desirable properties of the output by using the fitness function. The operators of a GA (cross-over, mutation and tournament selection for example) then allows it to search for the closest match.

It should be noted that a GA is not necessarily required to achieve automation. The crime severity scale can be used as an input to a Datafly algorithm for instance to achieve k-anonymity. The average severity could be used to generate an AS_D-diverse dataset as well. However such implementations do not take information utility and therefore optimality into account. The GA approach enables us to combine all aspects into a single process which:

- Produces optimal results with respect to the privacy requirements (k-anonymity for CG-kanon and l-diversity for CG-diverse) and information loss as specified by LM_{CG}
- Automatically enforces a local recoding on the QIDs which further improves information utility

- Facilitates our attribute weighting scheme to provide flexibility from a third party's perspective in specifying the granularity of the QIDs

Our observation about possible biases introduced into the data by the MCRF proved to be an interesting aspect of our work. We did find, however, that adding yet another parameter into the fitness function of our GAs to counter possible biases degraded the quality of our optimisations. Therefore whilst we considered possible measures to identify and correct for biases the practicalities for GAs may need some further work. For completeness we included an alternative measure for the adjustment factor which we derived. This is shown in Appendix B. However pilot experiments proved that our earlier definition for the adjustment factor in Section 4.8 was more effective to compensate for biases in our GAs.

We now turn to consider our results as based on the implementation which was covered in this chapter.

Chapter 5

Results

This section presents our experimental findings in support of earlier discussions from Section 3 and Section 4. We demonstrate the feasibility of our automated anonymisation scheme for crime data using a genetic algorithm given our resource constrained environment within a developing country. The three constraints faced relate to human capital (data anonymisation expertise) within the law enforcement agency, computational capacity and time constraints.

We have divided our results into two parts. Section 5.1 covers points relevant to our experimental process. Genetic algorithms can and have been applied to a broad range of problems. To harness the potential of GAs for practical purposes often requires an iterative approach to construct suitable implementations. This can possibly be seen as the downside of the flexibility provided by a GA. Our tests undertaken to find a suitable implementation are therefore quite informative to understand our final implementation and results. Section 5.2 evaluates these final results and demonstrates to what extent our initial objectives in this paper have been met.

5.1 Experimental process

5.1.1 System specifications

Hardware specification of the host server was as follows: Ubuntu server 12.04 operating system running on a 64bit machine with 8GB RAM. Processor speed of 3.2GHz (Intel Xeon E3-1230 Quad Core). Java 1.7.0_65 was installed to run the GA and Python 2.7.3 was used to run the Flask-based web server. PostgreSQL 9.1 was selected as our database management system. Postfix was installed as our email server. The entire implementation was run on a rented

dedicated server with a static IP to facilitate the web application.

5.1.2 Preliminary work

Solving the anonymisation problem with a GA brought with it the following considerations and challenges:

- A core GA framework would need to be selected as the complexities of building our own is beyond the scope of this paper
- The multi-objective nature of the anonymisation problem needs to be catered for by the above framework
- The computational complexities of GAs require significantly more time to test and obtain results. Anonymisation time for traditional implementations of k-anonymisation algorithms such as Datafly, Mondrian and Incognito can be measured in milliseconds for moderate sized datasets. The equivalent can take several minutes or even hours for larger datasets using GAs
- GA optimisation caters for flexible problem specification, however with this come difficulties in determining appropriate inputs and parameters

5.1.2.1 An appropriate framework

The requirements for a suitable framework were summarised as follows:

- It needs to be cross-platform to facilitate adoption across various law enforcement agencies and/or departments
- The framework and programming language implementation had to support the multi-objective nature of our anonymisation problem
- A modular framework is preferred as this ensures flexibility to structure our implementation and provides scope for modifications going forward if required
- A well documented framework is critical due to the inherent complexities of GAs

Given the above requirements the Opt4j framework[1] of [37] was chosen for the GA component of our implementation. The framework is written in Java. This facilitates cross-platform implementation and provides a good threading model for multiprocessing suited to reducing runtime

[1]Available at: http://opt4j.sourceforge.net/

for multi-objective optimisation problems such as ours. Evolutionary algorithms (EA), as a subset of GAs, have been widely applied to multi-dimensional problems. An appropriate selector is however required for multi-objective EAs. Opt4j included a range of selector implementations including two of the most popular namely, SPEA2 and NSGA2.

The modular nature of the framework is evident from Figure 3.2 where Opt4j is at the core of the CG anonymisation algorithm. The modularity is made accessible by well documented source code and informative tutorials on how to combine various components. According to the authors 51% of the source code are comments. Two other Java frameworks for GAs which might be explored by interested readers are the *Watchmaker framework*[2] and *EvA2*[3].

5.1.2.2 Formulating a practical implementation

Computational and time complexities associated with GAs meant that various aspects had to be considered before implementation. This was to ensure that our final solutions are practical and can provide satisfactory results given our problem setting. We look at salient points related to this in the discussion to follow.

- **Record size.** Our generated dataset consisted of 10000 records. We selected much smaller samples from this set to monitor computation times. A simple fitness function was specified which simply required 5-anonymity. No loss metrics or penalty functions were included at this stage. The algorithm stopped once all equivalence classes met 5-anonymity. Note that this is different from our final implementation and served only as a baseline to monitor the impact of record size on a GA. We can see from Figure 5.1 that firstly most of the benefit is derived early on and secondly that the marginal benefit decreases over time. This agrees with findings in other works employing GAs.

  A subsequent run was conducted on the full dataset of 10000 records where the fitness function for CG-kanon was used. A logging module outputted the progress of the GA after each iteration - this optimisation log was monitored. The process was stopped after 5 hours where roughly 200 equivalence classes still did not meet the CG-kanon requirement. Although progress had been made by the algorithm this was clearly too slow given our problem setting.

  The points to follow discuss measures used to address this runtime issue. We show that with some minor adjustments to our implementation we can reduce computation time whilst still benefiting from the GA approach.

[2]Available at: http://watchmaker.uncommons.org/
[3]Available at: http://www.ra.cs.uni-tuebingen.de/software/JavaEvA/index.html

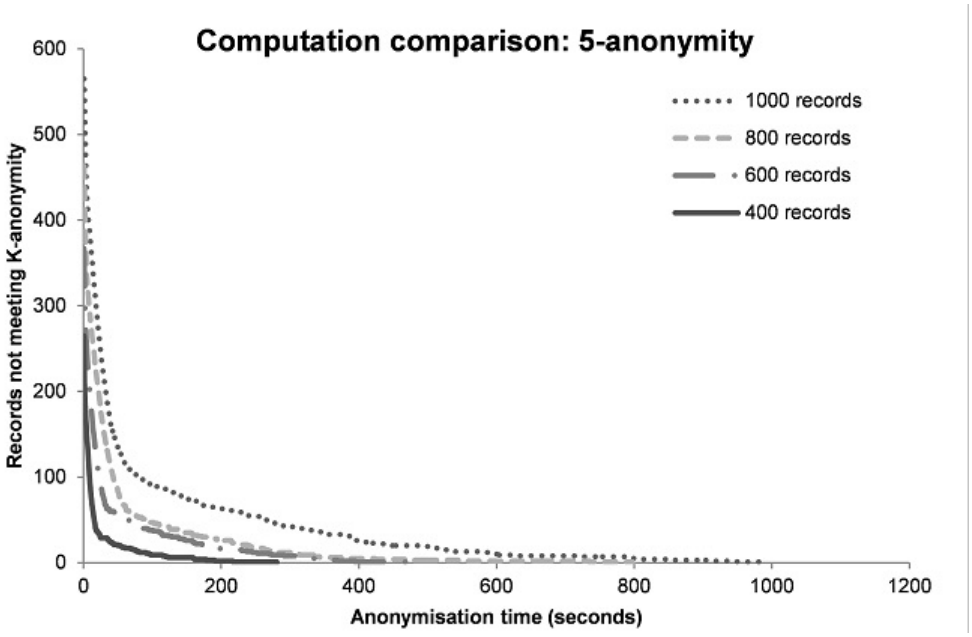
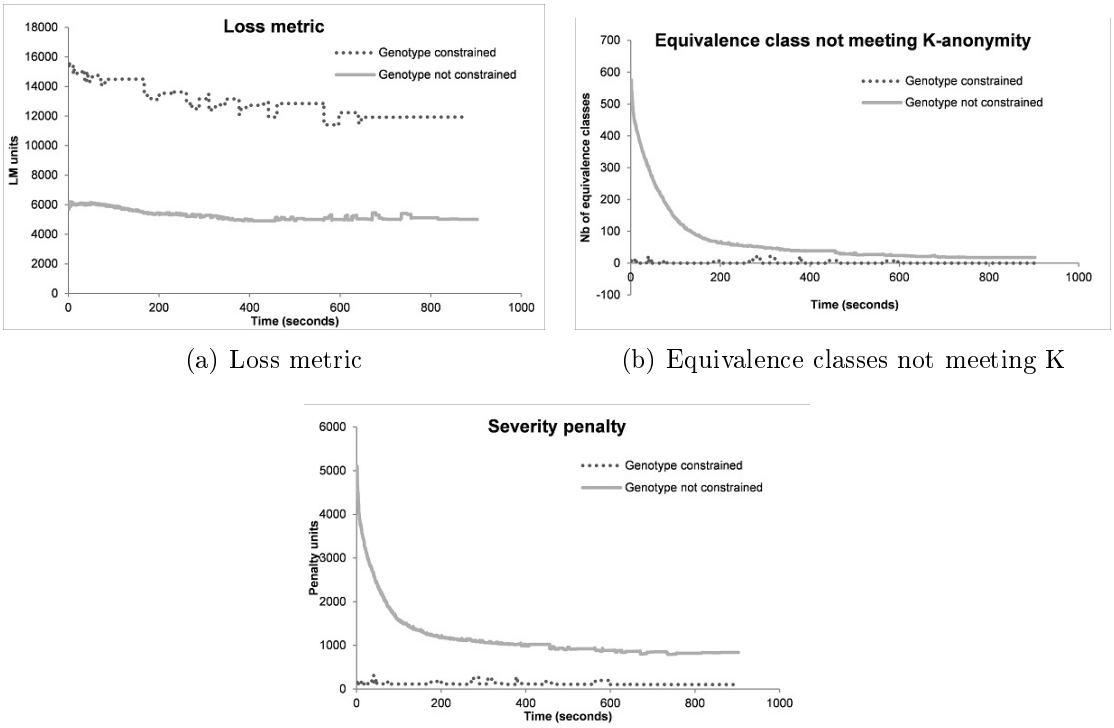


Figure 5.1: Impact of record size on computation time

- **Sampling.** As a first step we reduced the number of records processed for anonymisation - we used the idea of sampling 1000 records at a time for each data request. As noted in our literature review the sampling of records however introduces new challenges. In our particular case, the QID weight selection from Section 4.4.3 created additional disclosure risks since users of the data are given more flexibility to specify features of the data. We selected sampling without replacement as a simple solution to this problem in Section 4.7.2.1.

- **Pre-processing.** We refer to pre-processing as modifying the genotypes of the GA before starting the optimisation process. As described in Section 3 the genotype defines the building blocks accessible to the GA for constructing a solution. We considered selectively generating genotypes which more closely match our main priority during anonymisation. More specifically we forced genotypes to meet the k-anonymous requirement for CG-kanon before entering as candidates into the starting population. This was done by generalising attributes to the highest node for equivalence classes that do not meet the k-anonymity requirement.

  Figure 5.1.2.2 shows our experimental output for CG-kanon which was run for 15 minutes. Figure 5.2(b) and Figure 5.2(c) show that we obtain a significant initial benefit for equivalence classes not meeting k and the severity penalty respectively through pre-processing. The reduction in the severity penalty is derived from larger equivalence classes produced when pre-processing for k-anonymity.



(a) Loss metric

(b) Equivalence classes not meeting K

(c) Severity penalty

Figure 5.2: Impact of pre-processing on anonymisation metrics

However, the benefit for these two parameters is soon eroded after only a few minutes due to the exponential improvement shortly after optimisation. The stickiness of the loss metric in Figure 5.2(a) is now of greater concern. This implies that we have over-generalised significantly when the genotype is constrained. Another interpretation of this result is shown by Figure 5.3.

We see that Figure 5.3 confirms the over-generalisation. Firstly we note that pre-processing resulted in anonymised records distributed over only 18 equivalence classes[4]. Non-constrained genotypes created 111 equivalence classes (after the 15 equivalence classes not meeting k_{min} were generalised to the highest node, i.e. rolled into $G18^*$).

The above results informed our decision that the information loss of fully generalising a small number of equivalence classes after a pre-determined runtime outweighs the cost of over-generalisation due to pre-processing. Pre-processing does however enforce a stricter privacy throughout the optimisation process and might be considered where we want to enable the user to interrupt the optimisation at any stage and retrieve the data as is. An improved pre-processing module might also be considered to retain more information during this stage. Applying a modified Datafly algorithm for instance to enforce the k-

[4] $G18^*$ represent the equivalence class where the QIDs are at the highest generalisation node, i.e. 18_87;Cape Town;Reporter

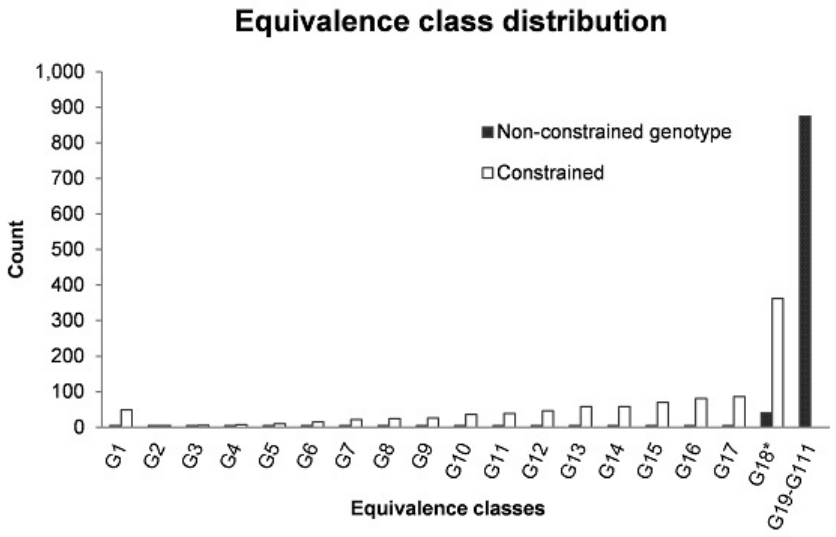


Figure 5.3: Impact of pre-processing on equivalence classes

anonymity prior to entering the GA might be a solution. However the modified Datafly algorithm should generate sufficiently random generalisations to qualify as input for a GA. Furthermore, since CrimeGenes uses a local recoding model to improve information retention this would somehow need to be accounted for in the pre-processing as Datafly is based on a global recoding model.

Currently the proposed pre-processing over generalises by removing too much detail and requires significant processing time to recover from that. Our results showed that suppression after anonymisation is much more effective and the desired loss metrics and severity penalties can be optimised more successfully.

- **Restricting the search space.** The ability of EAs to solve multi-dimensional problems is derived from the fact that the entire solution space is effectively explored by the algorithm. As alluded to however in one of our earlier points this can result in significant computation times because irrelevant or impractical solutions are explored.

  Restricting the search space can be achieved firstly through redefining the range of acceptable variables before optimisation or secondly by guiding the optimisation process itself. In our setting the sample data consists of two QIDs with highly granular generalisation hierarchies, namely Age and Suburb. This increases the search space and therefore computational complexity. We could consider pruning generalisation hierarchies to reduce computational complexity if other efficiency measures as discussed in this section fail. However, as mentioned in Section 4.1.1 using granular generalisations can be viewed as a worst-case scenario for testing our implementation. In short, proving the validity of results for sub-optimal data is preferred to using data suited to the specific implemen-

tation. We therefore did not change the generalisation hierarchy but instead looked to target the optimisation process itself.

For CG-kanon in particular, we sought to guide the EA in its search for optimal solutions by inducing a penalty to the fitness function if:

1. It explored solutions resulting in large numbers of equivalence classes not meeting the level of k-anonymity
2. It created equivalence classes larger than 10% of the total records in the sample

Note that for neither of these we stated a specific level or constraint, we simply guided the process to remain bounded to a location in the search space we view as desirable. As an example observe in Figure 5.2(b) how the dotted line is allowed to deviate briefly from zero but promptly returns to the x-axis. This same technique assists in rapidly improving the unconstrained genotype in the same figure.

- **Algorithm inputs.** In addition to setting parameters for the fitness function we need to set the population size, number of iterations , the number of offspring and cross-over rates for the GA as a whole.

  One complication when using a Pittsburgh approach is that each individual on its own comprises a solution to the problem. This increases demand for memory as mentioned in Chapter 2 during our review of the literature. Population sizes exceeding 200 caused the Java Garbage Collector (GC) to throw an error as the memory limit was reached. We experimented by manually expanding the memory allocated to the GA process. Although increased memory allocation was a possible solution we settled on decreasing the population size to 100 instead. After sufficient experimentation we set the above input parameters as follows:

  - Starting population at 100
  - Number of offspring at 100
  - Cross-over rate was set at the default Opt4J level of 0.95
  - Number of iterations were set at 1000000. However note that this would never be reached as our stopping criteria (discussed next) would terminate the GA well in advance. This parameter was therefore set sufficiently large as to never be reached

- **Stopping criterion.** We discussed relevant stopping criterion for GAs in our literature review. Due to the open-ended nature of the optimisation process for GAs it remains difficult to determine in advance when the optimisation will complete. From a practical perspective, however, users will require some guarantees in this regard. Results from our

preliminary tests demonstrate that improvement of results are exponential shortly after initialisation but fairly constant after large number of iterations. We therefore looked for a compromised approach which exploits the exponential improvement early on and gives users a result within a reasonable amount of time.

Our experiments in this regard revealed that after 30 minutes between 3% and 6% of tuples were not in equivalence classes meeting the minimum level of k for CG-kanon. This was weighed against running the algorithm for much longer than 30 minutes to ensure full compliance with the level of k. Since 3% to 6% is relatively small we opted to stop the algorithm after 30 minutes and generalise such outstanding tuples to the highest level. We note the following two points in relation to this:

1. Figure 5.3 shows that with pre-processing the number of tuples suppressed was as high as 35%. Therefore although the pre-processing resulted in termination of the algorithm without requiring us to set a limit on runtime, the information loss of suppressing such a large percentage of tuples was deemed too significant. A more efficient pre-processing algorithm might enable us to reconsider this approach if a limit on runtime is considered undesirable
2. Tuples in equivalence classes not meeting the level of k-anonymity will typically relate to less severe crimes due to the crime severity penalty of CrimeGenes. Generalising such tuples to the highest level will help alleviate the inferential attack described earlier where large equivalence classes have too little diversity.

Our discussion in this section has hopefully provided the reader with a useful background when evaluating our actual results. We now turn to these results which incorporate the issues highlighted above.

5.2 Experimental evaluation

As outlined in the previous section our final implementation was informed by extensive experimentation. Our results and interpretations thereof follow below. We begin this section by qualitatively assessing the anonymised data produced by the CG-kanon model and the CG-diverse model introduced in Chapter 4. We then consider the information loss generated by these two models before looking at the classification accuracy and KL-divergences of the two approaches. Our results also show the impact of introducing our QIDs weighting scheme and we comment on this as appropriate.

Throughout the results we will refer to an anonymisation based on the weightings of the QIDs used during the anonymisation. This will be denoted as $Aw_{Age} : Sw_{Suburb} : Rw_{Reporter}$. For example where equal weights were assigned to the QIDs this will be denoted as an A1:S1:R1 anonymisation; similarly where we use A10:S5:R1 weights of 10, 5, and 1 were used for the *Age*, *Suburb* and the *Reporter* attributes respectively.

$k_{constant}$ was set to 5 for all results on CG-kanon anonymisations. Our minimum crime severity level for the data was 3. Looking at our definition for k_{min} we see that this implies k_{min} is equal to 5. For CG-diverse our lowest level achieved for diversity was 3 across all anonymisation runs. Since on average the lower severity crimes were located in such equivalence classes this was acceptable. We will discuss this again below.

All algorithms were allowed to run for 30 minutes after which the algorithm was stopped. Once stopped the anonymised data was checked for compliance with the desired level of privacy. Tuples not satisfying the privacy criteria on termination were processed further according to the respective CG-kanon or CG-diverse algorithms in Section 4.9.

5.2.1 Qualitative assessment

In this section we look in particular at the distributions of the anonymised data provided by both CG-kanon and CG-diverse to determine whether our GA has in fact anonymised the data as expected.

Figure 5.4 shows an extract from an anonymised dataset using CG-kanon. Whilst the privacy definitions vary between the two approaches, we note the following points which are common to both when looking at the sample:

- A local recoding result is achieved which means that tuples with the same QIDs can be generalised differently across the entire dataset. A sample dataset in Figure 2.2 showed how this can improve data utility
- The reported crime is not suppressed even where all other QIDs are suppressed. Although this adds further utility to the anonymised data (third parties can derive the empirical distribution for reported crimes) it makes the anonymised data for susceptible to inference attacks

We now turn to analysing the distribution of the sensitive attribute after applying CG-kanon and CG-diverse. Analysing this distribution gives us insight into how well privacy risk is addressed by the algorithms. Information loss and utility metrics later deal with the opposing question about data utility.

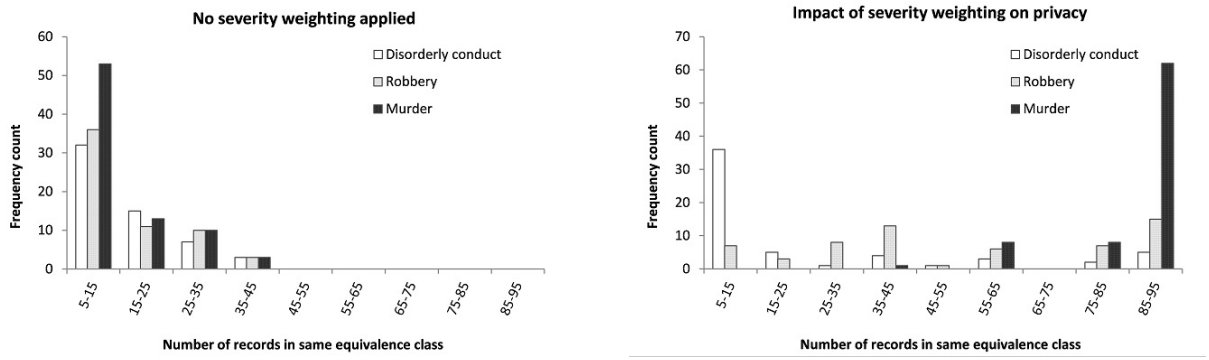
ID	Firstname	Lastname	Gender	Age	Address	Suburb	Region	Cell number	Reporter type	Occurrence	Time	Crime type
3	Alison	Cameron	F	31	Fairway Drive	Camps Bay	Atlantic Seaboard	842820568	Witness	Oranjezicht	6	Assault
4	Alan	Young	M	47	Central Avenue	Fresnaye	Atlantic Seaboard	800297285	Witness	Bo-Kaap (Malay Quarter)	7.5	Vandalism
5	Chloe	May	F	49	Sycamore Street	Walmer Estate (District Six)	City Bowl	800906044	Victim	Higgovale	11	Family or Domestic Violence
6	Christian	Terry	M	39	Church Road	Kraaifontein	Northern Suburbs	889825750	Proxy	Bothasig	19.5	Disorderly conduct
7	Karen	Anderson	F	32	Devonshire Drive	Goodwood	Northern Suburbs	829100177	Victim	Bo-Kaap (Malay Quarter)	20.5	Assault

ID	Firstname	Lastname	Gender	Age	Address	Suburb	Region	Cell number	Reporter type	Occurrence	Time	Crime type
3	*	*	*	18_87	*	Atlantic Seaboard	*	*	Witness	*	*	Assault
4	*	*	*	43_47	*	Cape Town	*	*	Witness	*	*	Vandalism
5	*	*	*	18_87	*	Walmer Estate (District Six)	*	*	Reporter	*	*	Family or Domestic Violence
6	*	*	*	18_87	*	Northern Suburbs	*	*	Proxy	*	*	Disorderly conduct
7	*	*	*	18_87	*	Cape Town	*	*	Reporter	*	*	Assault

Figure 5.4: Sample anonymisation

5.2.1.1 CG-kanon

The size of an equivalence class is the only privacy tool at our disposal within a k-anonymous model. Our results show the impact of moving more severe crimes to larger equivalence classes using the CG-kanon algorithm. Figure 5.5 shows the distribution of crime reports with different severities after applying CG-kanon with a A1:S1:R1 weighting. We see in Figure 5.5(a) that without the CG-kanon severity penalty the three types of crimes reported are clustered around smaller sized equivalence classes. Where the severity penalty is applied however we see in Figure 5.5(b) that more severe crimes (Robbery and Murder in this case) are located in larger equivalence classes. In general the crime reports are also more spread out across equivalence classes. Whilst the algorithm performs as expected this introduces unwanted inferential disclosure.



(a) No severity weighting

(b) CG-kanon

Figure 5.5: Impact of introducing severity weighting

The following privacy concerns arise when one breaks down the constituents of the larger

equivalence classes:

- Larger equivalence classes by definition contain more severe crimes. This alone discloses some information
- The distribution of reported crimes has a significant impact on how serious the problem above can become. Figure 5.6 gives a breakdown of the largest category (85-95) from Figure 5.5(b). For this subset of anonymised data under CG-kanon we see that the more severe crimes have a higher frequency with *Murder* being as high a 31%. This is the equivalent of a 3-diverse dataset and provides inadequate inferential privacy protection given the severity of such a crime. Figure 4.2(c) showed that *Murder* had the highest frequency for a single offence in our generated data. This contributed to the problem seen in these larger equivalence classes under CG-kanon. If for instance the frequency of *Murder* was lower this would have resulted in more diversity.

	Equivalence classes			Crime severity
	18_87Cape TownReporter	18_87Cape TownVictim	18_87Cape TownWitness	
Arson	7%	8%	2%	10
Assault	2%	9%	6%	7
Burglary	1%	1%	2%	5
Corruption or Embezzlement	8%	1%	1%	3
Disorderly conduct	4%	1%	0%	3
Drug related	**12%**	**10%**	**10%**	**10**
Drunken Driving	8%	0%	1%	5
Family or Domestic Violence	5%	0%	1%	5
Forgery or Fraud	12%	9%	11%	15
Illegal gambling	4%	1%	3%	5
Murder	**12%**	**31%**	**27%**	**25**
Other	3%	5%	4%	5
Rape	**9%**	**16%**	**19%**	**20**
Robbery	4%	7%	6%	7
Theft	2%	1%	7%	5
Vandalism	5%	0%	0%	3

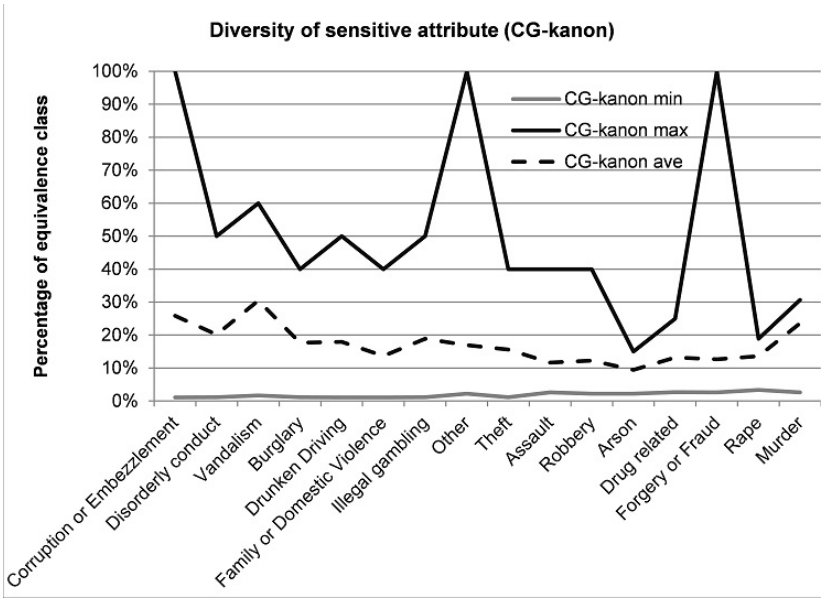
Figure 5.6: Shortcoming of CG-kanon

We note that whilst the combination of k-anonymity with a severity penalty seems quite straightforward we should recall that this is done whilst minimising information loss and applying the desired weightings for the QIDs. The implementation of this is therefore rather more complex and served as a valuable stepping stone to develop the more robust CG-diverse GA to which we now turn our attention.

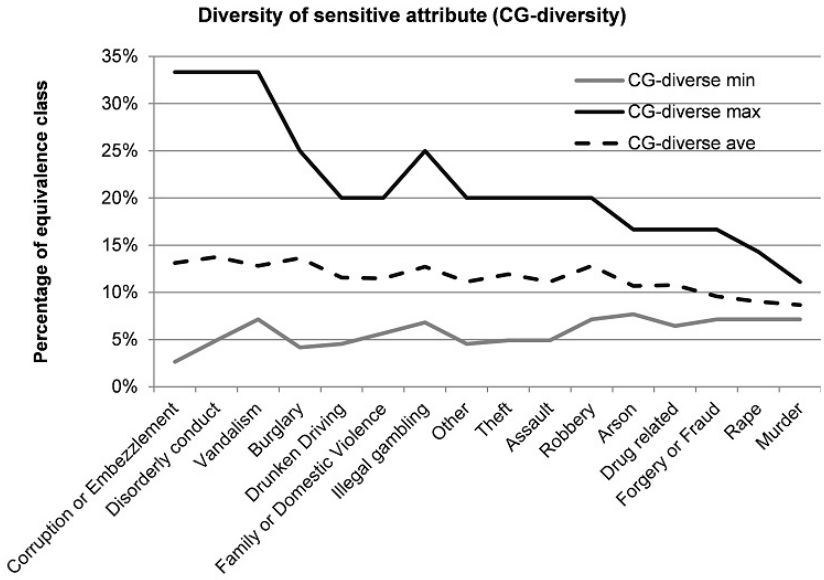
5.2.1.2 CG-diverse

As discussed in Chapter 4, CG-diverse is based on the notion of l-diversity. Our results show that this stronger notion of privacy can be implemented by a GA and that the shortcomings

of CG-kanon can be addressed by CG-diverse. Our results here are based on the A1:S1:R1 weighting however results for other weightings are similar. The average severity level obtained for the data used in our experimentation was 11. As noted, however, in Chapter 4 the second phase of the algorithm reduces this global diversity requirement and considers the average severity of each equivalence class before suppressing the QIDs. The lowest such diversity level within an equivalence class was 3 in our results.



(a) CG-kanon



(b) CG-diverse

Figure 5.7: Sensitive attributes frequency for CG-kanon and CG-diverse using A1:S1:R1

We comment on the following points when looking at Figure 5.7:

- **Higher diversity (lower frequency).** Average diversity for CG-kanon varies between 10% and 30%. CG-diverse is much lower at 9% to 14% and therefore inferential risk is also lower
- **Correlation with crime severity.** The desired lower frequency (i.e. higher diversity) for more severe crimes is evident in CG-diverse whereas in CG-anon there is no such correlation. More severe crimes (Rape and Murder) in this case actually have lower average diversity
- **Variation of diversity.** We see the deviation from the mean frequency for more severe crimes is lower as severity increases. So not only does the average diversity increase as crime severity increases but the variance decreases as well. This gives us more certainty that more severe crimes will be less vulnerable to inference attacks
- **l-diversity implies k-anonymity.** We mentioned before that l-diversity guarantees at least k-anonymity where k is equal to l. The lowest diversity of 3 mentioned earlier for our dataset using CG-diverse may appear quite weak from a privacy perspective when looking at the global diversity of 11. However, it is firstly much more unlikely that severe crimes will be included in such lower diversity equivalence classes in CG-diverse. Secondly, we are still better off that in CG-kanon. Our results there showed that even for severe crimes such as *Murder* the largest equivalence classes still only achieved 3-diversity in some cases.

5.2.2 Optimising for utility

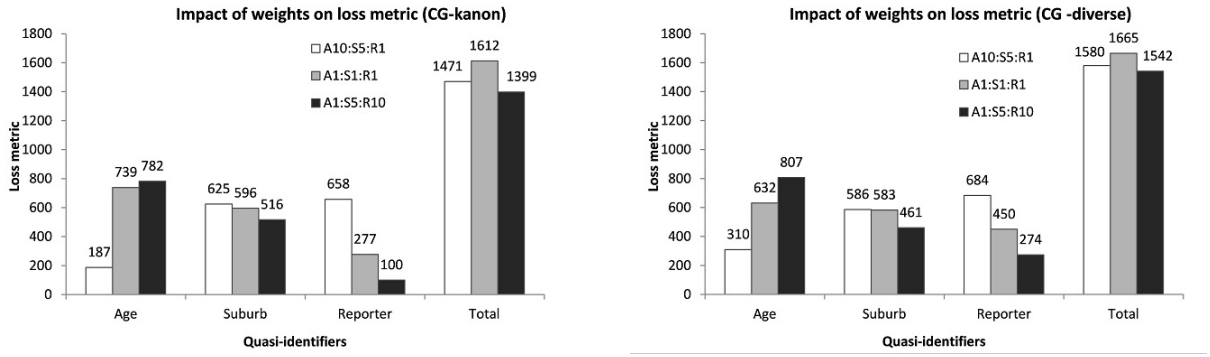
5.2.2.1 Information loss

The losses assigned to generalised attributes within a tuple were set out in Section 4.4. Figure 5.8(a) and Figure 5.8(b) show the aggregated information losses for different weighting schemes after termination of the algorithm.

We selected the three weighting schemes to monitor how the algorithms perform when attributes with varying granularity are weighted differently. For instance the A10:S5:R1 scheme overweights the *Age* attribute which is highly granular and underweights the *Reporter* attribute with only 3 leaves - A1:S5:R10 tests the opposite scenario. A1:S1:R1 however is equivalent to having no weighting scheme.

We make the following observations about the losses and weighting schemes:

- We see the weighting scheme achieves the desired loss profiles. Where *Reporter* is prioritised for anonymisation (A10:S5:R1) we see more information loss for that attribute



(a) CG-kanon

(b) CG-diverse

Figure 5.8: Loss metrics for CG-kanon and CG-diverse

- the converse is true for A1:S5:R10. Where no weighting is assigned variation of losses decreases

- Total information loss for CG-diverse is slightly higher than CG-kanon. This is expected due to the stronger privacy guarantees of CG-diverse
- Total loss for A1:S5:R10 is lowest for both algorithms. The tendency for the algorithm to suppress the more granular *Age* attribute under this weighting causes this
- The attribute weighting for CG-diverse is still effective, however, less so than for CG-kanon. For instance, we observe a lower variation between the maximum and minimum loss for A10:S5:R1 and A1:S5:R10 in CG-diverse compared to CG-kanon. The restriction on diversity is prioritised and thereby the weighting is less significant during optimisation

Appendix C shows sample anonymised data from CG-kanon and CG-diverse where crime reports with the same report IDs were selected. The output confirms the impact of the attribute weighting scheme as measured by the loss metric above. By inspecting the sample output from CG-kanon we see the following for example:

- **A10:S5:R1**. There are 6 tuples where age is left unchanged whilst the reporter type attribute has 24 tuples that were generalised to the highest level ("Reporter")
- **A1:S5:R10**. There are 36 tuples where age was generalised to the highest level ("18_87") whereas the reporter type attribute now only had 3 tuples generalised to the highest level ("Reporter")

Our sample output from CG-diverse as shown has similar characteristics. However the differences are less pronounced. This is confirmed by what we see when the loss metrics are compared.

5.2.3 Utility metrics

The Weka framework was used to perform classifications on different anonymised datasets. The anonymised datasets were generated by applying different anonymisation algorithms to the same sample of crime reports (sample size of 1000 records). The results from a 10-fold cross-validation using a Naive Bayes inducer are shown in Figure 5.9. We performed classifications on both anonymised datasets and their corresponding regenerated counterparts (i.e. we took the anonymised data and randomly generated non-anonymised data from this).

5.2.3.1 Classification

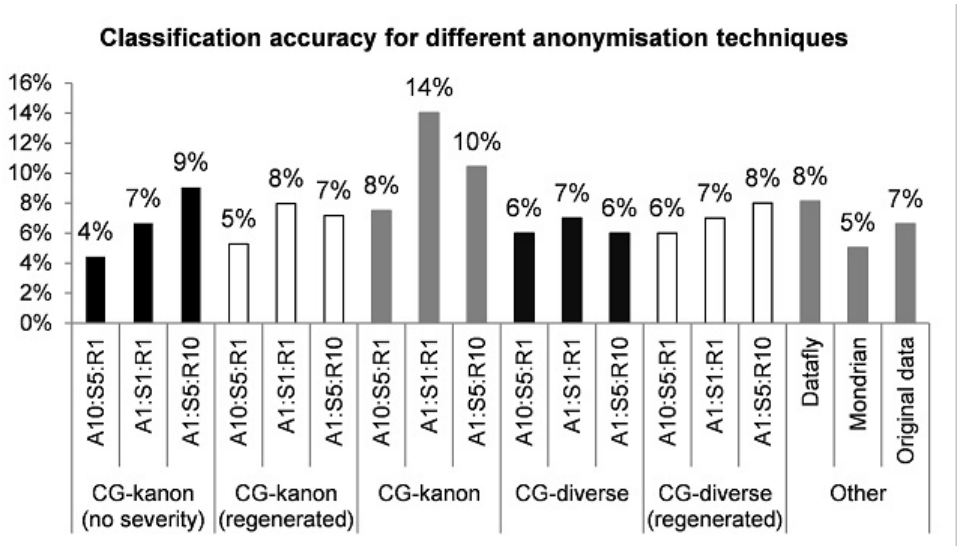


Figure 5.9: Classification accuracy for different anonymisations

We see from Figure 5.9 that all classifications have very low accuracies. This was alluded to in Section 4.1.1 where we pointed out that crime categories were randomly assigned to tuples. We should therefore not expect much classification accuracy. The validity of using classification accuracy as an indicator for the utility of an anonymisation is case specific. Where the raw data has low classification accuracy one should not look for high accuracies in the anonymised data.

The relative higher classification accuracy of CG-kanon is a symptom of the shortcoming mentioned earlier. More severe crimes are located in larger equivalence classes. This clustering induced by the severity penalty in CG-kanon improves classification accuracy. It is interesting to note the lower accuracy for the corresponding regenerated dataset. This affirms our discussion earlier in Section 2.4 which indicates that where classification is conducted it should

be done on the regenerated (non-anonymised) data to avoid spuriously induced classification accuracy.

One observation we might make is that prioritising the anonymisation of the more granular *Age* attribute (A1:S5:R10) results in slightly higher classification accuracy than when the *Reporter* attribute is prioritised for generalisation (A10:S5:R1). This would support the notion in [7] where better classification accuracy was achieved by prioritising the numerical age attribute.

5.2.3.2 Kullman-Leibler divergence

A script in Python was written to calculate the Kullman-Leibler(KL) divergence for various algorithms. Figure 5.10 indicates the variation in the KL-value if various proportions of the original (non-anonymised) data are suppressed. This also served to validate our script.

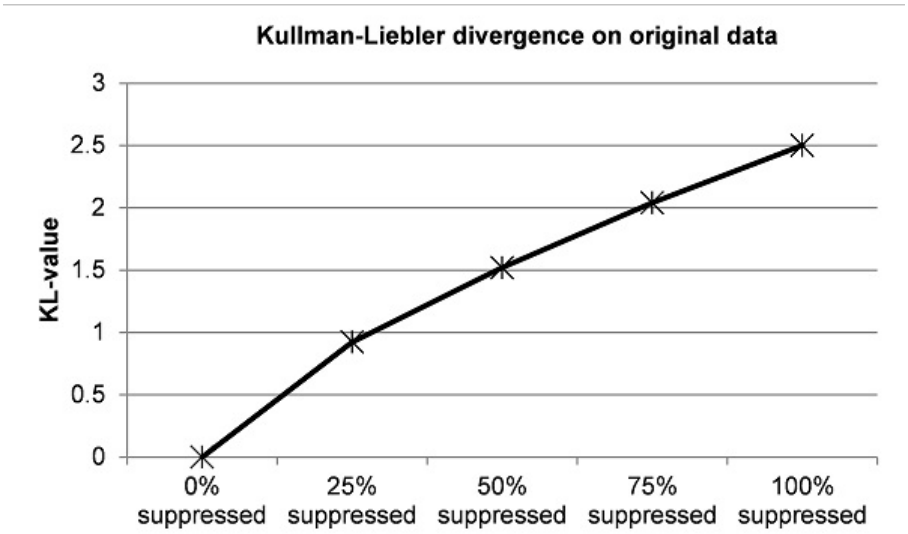


Figure 5.10: KL-divergence at various suppression levels

Where we have highly granular attributes one should expect anonymised data based on generalisation to have fairly high KL-values. This is evident from our results in Figure 5.11.

Analogous to our results for information loss, the KL-divergence for CG-diverse is slightly higher for all weightings compared to CG-kanon due to the higher privacy guarantee. The benefit of using KL-values is, however, that it provides a standard means for comparing different anonymisation algorithms or anonymisations with different loss metrics. The statistical interpretation of the KL-divergence as covered in Section 2.4 also provides a firmer foundation for this metric.

The impact of the weighting scheme on the KL-value is lastly worth commenting on. In contrast

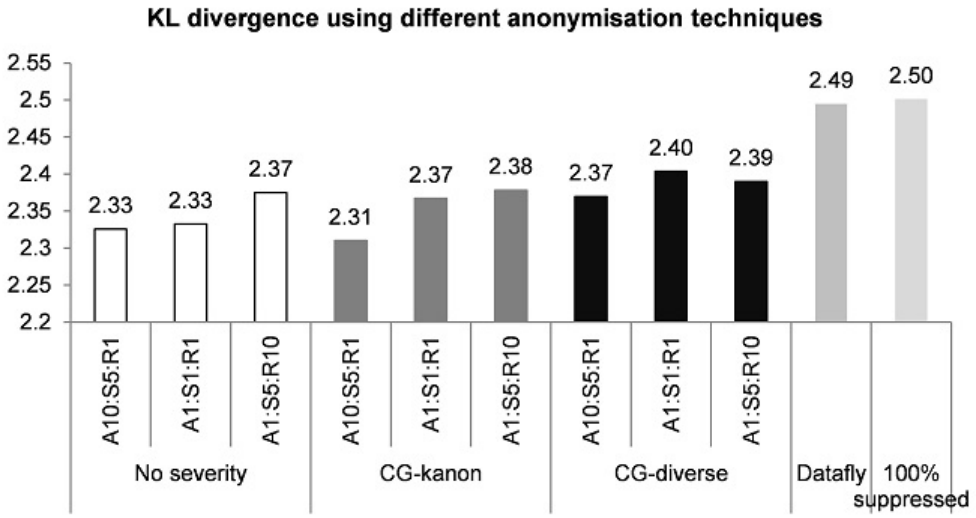


Figure 5.11: KL-divergence for different anonymisations

to the classification accuracy a lower KL-value is obtained when more granular attributes are generalised to a lesser extent. More granularity is retained in the data for A10:S5:R1 compared to A1:S5:R10. On this basis we would prefer A10:S5:R1 if looking at KL-divergence. This is the opposite conclusion reached when looking at classification accuracy.

The marginal increases in information loss and KL-values for CG-diverse relative to CG-kanon seem quite acceptable given the improved privacy provided by the former. For our results the information loss across the three weighting schemes was on average 7% higher and the KL-value only 1.4% higher for CG-diverse compared to CG-kanon. This reduced data utility is acceptable given our desire for better privacy within the MCRF.

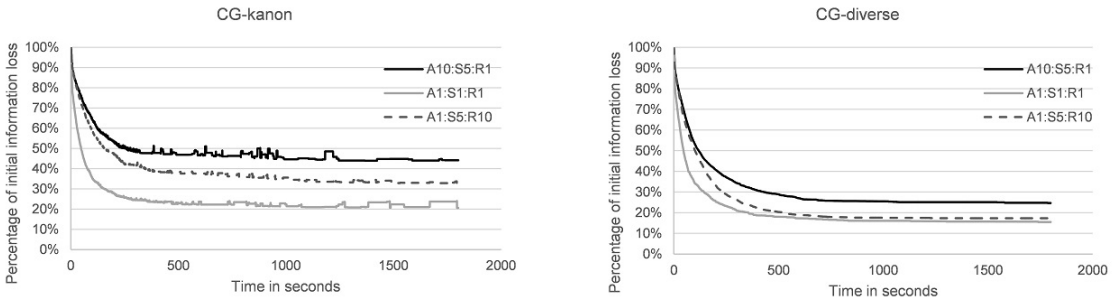
5.2.3.3 Computational and time constraints

Computing power and time constraints dictated much of our implementation. Section 5.1 already highlighted some relevant issues.

The hardware platform for our implementation was an entry-level dedicated server hosted remotely by a third party. In practice the law enforcement agency may prefer a local server to restrict physical access to it. The specifications and hardware utilised for our implementation were therefore not excessive and should be easily obtainable and implementable.

Operating within the time constraints was however more challenging. The open-ended nature of GAs required us to define a time limit per anonymisation to make the implementation practical. Figure5.12 shows the reduction for information loss retrieved from the optimisation logs.

The marginal benefit from optimisation decreases exponentially over time. It was our initial judgement that 30 minutes allowed the algorithm sufficient time to improve the anonymisation whilst not letting the third party user wait indefinitely for a result. However when looking at Figure 5.12(a) and 5.12(b) we see that we could halve this time and still benefit adequately from the optimisation of the GA. Alternatively we might increase the sample size and keep the runtime of the algorithm at 30 minutes.



(a) CG-kanon

(b) CG-diverse

Figure 5.12: Information loss reduction versus time

One further insight we wish to point out relates to the number parameters which the GA optimises within the fitness function. We see that information loss for CG-diverse on termination is a much lower proportion of its starting value than for CG-kanon. This can be attributed to CG-kanon having more parameters in the fitness function than CG-diverse. CG-kanon searches for solutions that minimise information loss and the severity penalty whilst satisfying k-anonymity. CG-diverse only minimises information loss whilst meeting the diversity requirement. This one additional parameter for CG-kanon increases the search space and reduces the efficiency of the algorithm. For instance at termination the reduction in the initial information loss for A10:S5:R1 in CG-diverse was 74% compared to 55% for CG-kanon.

We lastly observe that the optimisation for information loss is much "smoother" for CG-diverse. This is again due to the simpler fitness function. The sudden peaks and troughs for CG-kanon occur as the algorithm introduces variation to improve optimisation. Figure 5.13 shows the severity penalty as well as the information loss optimisation for CG-kanon. A sudden increase in information loss often coincides with a drop in the severity penalty and vice versa. This additional dimension reduces optimisation efficiency and we see from this that whilst multi-objective EAs are suited to solving multi-dimensional problems there is a definite benefit when a simpler fitness function can be specified. This improved efficiency was evident for CG-diverse which has a simpler fitness function.

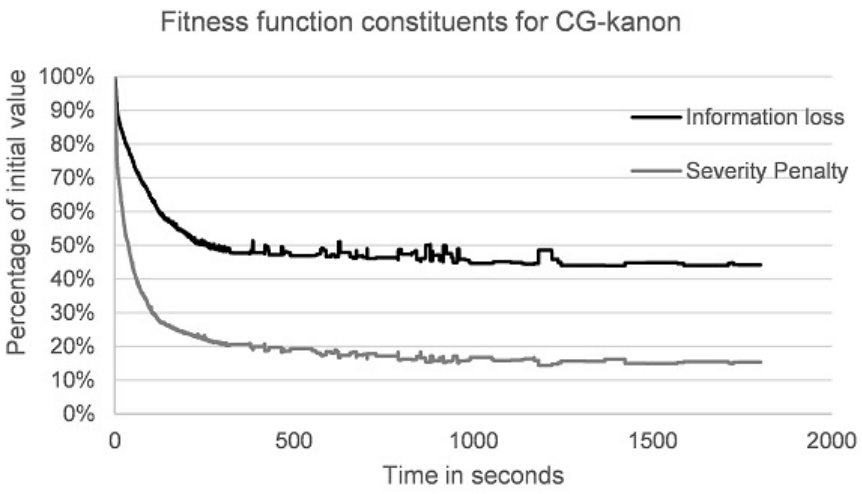


Figure 5.13: Optimisation parameters vs time

5.3 Discussion

We summarise our findings in this section and provide additional insights which may not be evident from our evaluation above.

A qualitative assessment of the distributions of the anonymised data show that CG-diverse is the preferred implementation. Whilst CG-kanon seeks to limit linking attacks it introduces unacceptably high inferential risk for more severe crimes in larger equivalence classes. Information loss is incrementally higher for CG-diverse but this is acceptable given the additional privacy guarantees against inferential attacks. We pointed out earlier that smaller equivalence classes were created for CG-kanon in an attempt to reduce inference attacks. Another possible approach which may be used is to consider all attributes as sensitive attributes as done by [7]. In this manner the reported crime attribute may also be suppressed which reduces distributional information relating to the reported crime.

When looking at the low classification accuracies and high KL-divergence values we reiterate that using suboptimal data (i.e. crimes were assigned randomly to tuples) prevents one from generating results designed to suit a specific utility metric. This was selected as a design feature of our experiments and therefore the somewhat unfavourable values of both utility metrics should not be seen as an interpretation of the success of CG itself. Values for the metrics produced intuitive metrics consistent with the rest of our results.

We specifically selected Java as the programming language to allow for threading during the optimisation process. This enables multiple processors to concurrently perform the optimisation

which can significantly reduce the runtime. The Opt4J framework was also selected with this in mind and provides documentation on how this may be implemented. Although we did not implement a multi-threaded CG optimisation we note that this may be considered in future to improve the runtime of the algorithm.

We summarise our results by observing that whilst the the shortage of human capital in our resource constrained setting was addressed by the GA combined with our crime severity scale, sampling had to be introduced to ensure that computational and time constraints were met.

Chapter 6

Conclusion

6.1 Summary

Our work has introduced the concept of a crime severity scale and shown how this might be applied to achieve automation for the anonymisation of data within the mobile crime reporting framework (MCRF) introduced by Burke and Kayem [7]. In particular we proposed a severity penalty to be used when anonymising data based on k-anonymity to reduce successful linking attacks; we also proposed an average severity measure used for an l-diverse approach to limit inference attacks.

We introduced an attribute weighting scheme that provides more flexibility for third party users in specifying the granularity of different quasi-identifiers (QIDs) in the anonymised data. This is achieved by modifying the original general loss metric introduced by [24]. We use a a genetic algorithm (GA) implementation to minimise information loss whilst achieving the required levels of privacy.

Our GA known as "CrimeGenes" (CG) is implemented based on both a k-anonymous model (CG-kanon) and an l-diverse model (CG-diverse). The GAs of both implementations integrate the entire anonymisation into a single process which achieves optimality with respect to information loss, the weighting of QIDs specified by the third party user and privacy constraints (k-anonymity or l-diversity informed by the crime severity scale). This is one of the main advantages of a GA approach - achieving optimal k-anonymity and l-diversity with respect to information utility whilst incorporating the flexibility to weight attributes differently. A local recoding of the anonymised data is also performed in our GA approach, although this can be done using more conventional algorithms as well. The main disadvantage of our GA approach compared to other anonymisation techniques is arguably the computational cost involved and

therefore the time it takes to generate the output. Most other conventional anonymisation algorithms (such as Datafly) are much faster but then lack the optimality and flexibility obtained by using a GA approach.

The computational complexity of the GA mentioned above required us to take a sampling approach to make our implementation practical given time constraints. However, to ensure privacy constraints are also met whilst making most recent crime reports accessible to third party users, we proposed a random sampling approach without replacement. A normal sampling approach which allows for replacement can increase disclosure risk. Crime reports released to a given third party in a previous data request are therefore excluded from being selected in subsequent releases. A sampling approach also reduced computational complexity and therefore runtime for our GA.

Our discussion above alludes to the fact that a crime severity scale in combination with a GA automates the anonymisation process and produces optimal results for a given runtime. This addresses the lack of expertise normally faced within a law enforcement agency in a developing country to anonymise data appropriately. Our sampling approach in releasing the anonymised data ensures our GA remains practical and can provide output to third parties within a reasonable time period. The computational and time constraints of our resource constrained environment are thereby also satisfied.

6.2 Avenues for future work

Our main focus for future work may relate to expanding on the notion of biases in the MCRF. We introduced possible means of capturing and compensating for such biases through an adjustment factor using our GA. Our preliminary work in this regard could be expanded to investigate the specific form of the adjustment factor further. Other existing techniques not employing a GA could also be modified to compensate for biases as part of the anonymisation process. Yet another approach might be to apply GAs as a pre-processing step on the raw data, not to perform anonymisation, but to identify possible biases and remove such records from the dataset. For instance, a GA might be used to identify multiple crime reports all reporting the same incident.

A focus area for continuous further improvement of our current CG implementation is the runtime of our algorithm. This is an issue relevant to GAs in general. Possible approaches which may be explored to decrease the runtime include:

- Employing a Michigan-based approach for the GA could be considered as discussed in

Section 2.5.3 and benchmarked against CG which uses the Pittsburgh approach

- Other optimisation techniques, such as particle swarm optimisation (PSO), which share similarities with evolutionary algorithms (EAs) may also be experimented with to see whether runtime can be improved. PSO is different in that it does not use the optimisation operators of cross-over and mutation common to most GAs
- The current version of CG could be developed further to make use of multiple processors. As noted in our results the current Opt4J (covered in Section 5.1.2.1) allows for threading although this complexity was beyond the current scope of our work

A last area of improvement relates to our sampling scheme. A more elegant approach may be investigated which allows either sampling with replacement or proper sequential releases as discussed in Section 2.3.7 of our literature review.

Appendices

Appendix A

UML for CG-kanon anonymisation engine

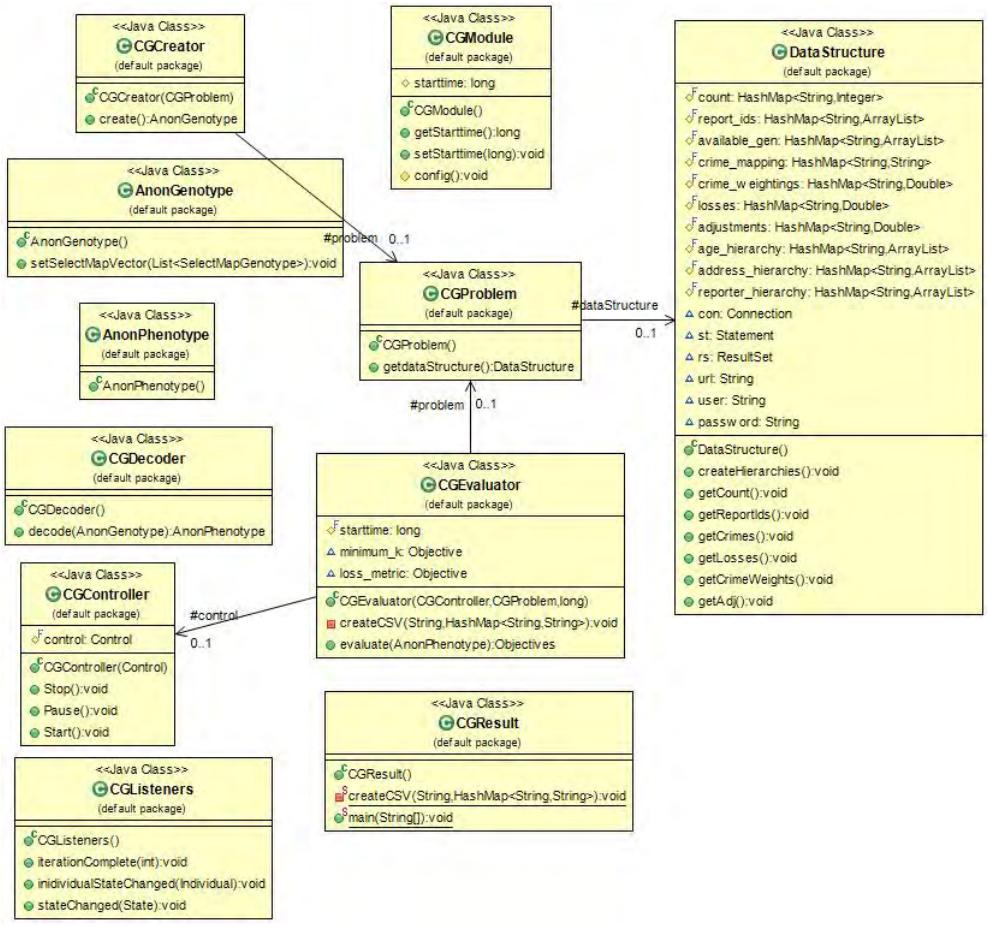


Figure A.1: UML for CG-kanon

Please note that the UML diagram for CG-diverse is exactly the same except that the fitness function of the GA differs as covered in Section 4.6.2.

Appendix B

Alternative adjustment factor

Definition B.0.1 (χ^2 cumulative distribution function)**.** The cumulative distribution function (CDF) for $Z \backsim \chi^2_n$ is defined as:

$$F(Z, n) = \frac{\gamma\left(\frac{n}{2}, \frac{z}{2}\right)}{\Gamma(\frac{n}{2})} \tag{B.1}$$

Where:

- $\Gamma(a)$ is the Gamma function
- $\gamma(a, b)$ is the lower incomplete Gamma function

Definition B.0.2 (The adjustment factor)**.** Since each crime category will be evaluated sequentially to obtain the adjustment factor (ad_c) for category c and since $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ and $\gamma(\frac{1}{2}, b) = \sqrt{\pi} \times \xi(\sqrt{b})$ equation B.1 can now be rewritten to define the adjustment factor for category c as:

$$F(Z_c, 1) = \frac{\gamma\left(\frac{1}{2}, \frac{z_c}{2}\right)}{\Gamma(\frac{1}{2})} = \frac{\sqrt{\pi} \times \xi(\sqrt{\frac{z_c}{2}})}{\sqrt{\pi}} = \xi(\sqrt{\frac{z_c}{2}}) \tag{B.2}$$

Where:

- $\xi(a)$ is the Gauss error function defined as $\frac{2}{\sqrt{\pi}} \int_0^a \exp^{-t^2} dt$
- This integral looks similar to the density function for a standard normal random variable and might lead us to consider employing the cumulative distribution function for a standard normal distribution to achieve a result. Indeed such a relationship exists

which enables us to rewrite this last equation using the standard normal CDF denoted as $\Phi(a) = \frac{1}{2} + \frac{1}{2} \times \xi(\frac{x}{\sqrt{2}})$.

Then:

$$F(Z_c, 1) = 2 \times \Phi(\sqrt{z_c}) - 1 \; for \; z_c \geq 0. \tag{B.3}$$

Finally:

$$ad_c = \frac{F(z_c, 1)}{\sum_{c \in C} F(z_c, 1)} \tag{B.4}$$

Since the standard normal CDF is widely used in practice efficient numerical techniques exist to evaluate this equation. This reformulation therefore enables us to quickly evaluate the adjustment factor for a specific crime category. However, as indicated in the main body of our work this formulation of the adjustment factor did not provide satisfactory results when incorporated into the GA.

Appendix C

Sample anonymised data

Table C.1: CG-kanon sample anonymised data

A10:S5:R1			A1:S1:R1			A1:S5:R10			CRIME
48_52	Southern Suburbs	Witness	48_52	Southern Suburbs	Reporter	18_87	Southern Suburbs	Witness	Assault
18_87	Cape Town	Reporter	18_87	Cape Town	Victim	18_87	Cape Town	Victim	Forgery or Fraud
_31	Cape Town	Reporter	18_87	Atlantic Seaboard	Witness	18_87	Atlantic Seaboard	Witness	Assault
43_47	Atlantic Seaboard	Reporter	43_47	Cape Town	Witness	18_87	Atlantic Seaboard	Witness	Vandalism
48_52	City Bowl	Reporter	18_87	Walmer Estate (District Six)	Reporter	18_87	City Bowl	Victim	Domestic Violence
38_42	Cape Town	Reporter	18_87	Northern Suburbs	Proxy	18_87	Northern Suburbs	Proxy	Disorderly conduct
28_32	Northern Suburbs	Reporter	18_87	Cape Town	Reporter	18_87	Northern Suburbs	Victim	Assault
28_32	Cape Town	Witness	18_87	Cape Town	Witness	18_87	Northern Suburbs	Witness	Forgery or Fraud
18_22	Cape Town	Victim	18_87	Cape Town	Victim	18_87	Cape Town	Victim	Rape
18_22	Northern Suburbs	Reporter	18_87	Durbanville	Witness	18_87	Northern Suburbs	Witness	Other
23_27	Cape Town	Witness	23_27	Southern Suburbs	Reporter	23_27	Cape Town	Witness	Disorderly conduct
_30	Cape Town	Reporter	28_32	Atlantic Seaboard	Reporter	28_32	Atlantic Seaboard	Witness	Illegal gambling
25	City Bowl	Victim	18_87	City Bowl	Victim	23_27	City Bowl	Victim	Other
23_27	Cape Town	Reporter	_26	Cape Town	Reporter	23_27	City Bowl	Victim	Illegal gambling
58_62	Cape Town	Reporter	18_87	Cape Town	Reporter	18_87	Northern Suburbs	Proxy	Murder
43_47	City Bowl	Reporter	18_87	Cape Town	Witness	18_87	Cape Town	Witness	Murder
18_22	Southern Suburbs	Reporter	18_87	Kreupelbosch	Reporter	18_87	Southern Suburbs	Witness	Robbery
28_32	Cape Town	Witness	18_87	City Bowl	Reporter	18_87	City Bowl	Witness	Domestic Violence
_45	Cape Town	Reporter	43_47	Cape Town	Witness	43_47	Cape Town	Witness	Burglary
23_27	Cape Town	Reporter	18_87	Cape Town	Proxy	18_87	Cape Town	Proxy	Arson
18_87	City Bowl	Witness	18_87	City Bowl	Witness	18_87	Lower Vrede (District Six)	Reporter	Drunken Driving
23_27	Cape Town	Reporter	18_87	Cape Town	Victim	18_87	Southern Suburbs	Victim	Murder
18_87	Cape Town	Victim	23_27	Northern Suburbs	Reporter	18_87	Northern Suburbs	Victim	Vandalism
18_87	Atlantic Seaboard	Witness	18_87	Cape Town	Reporter	78_82	Cape Town	Witness	Domestic Violence
18_22	Cape Town	Witness	18_87	Atlantic Seaboard	Witness	18_87	Atlantic Seaboard	Witness	Domestic Violence
23_27	Cape Town	Victim	18_87	Cape Town	Victim	18_87	Southern Suburbs	Victim	Forgery or Fraud
38_42	Northern Suburbs	Victim	18_87	Northern Suburbs	Victim	18_87	Northern Suburbs	Victim	Forgery or Fraud
23_27	Cape Town	Victim	18_87	Bantry Bay	Reporter	18_87	Atlantic Seaboard	Victim	Forgery or Fraud
43_47	Northern Suburbs	Proxy	18_87	Northern Suburbs	Proxy	43_47	Cape Town	Proxy	Domestic Violence
23_27	Cape Town	Proxy	23_27	Cape Town	Proxy	18_87	Southern Suburbs	Proxy	Assault
33_37	City Bowl	Witness	18_87	City Bowl	Witness	18_87	Cape Town	Witness	Assault
23_27	Cape Town	Reporter	23_27	Northern Suburbs	Reporter	23_27	Northern Suburbs	Reporter	Vandalism
18_87	Cape Town	Reporter	18_87	Northern Suburbs	Victim	18_87	Northern Suburbs	Victim	Domestic Violence
33_37	Atlantic Seaboard	Reporter	18_87	Atlantic Seaboard	Witness	18_87	Atlantic Seaboard	Witness	Drunken Driving
58_62	Cape Town	Reporter	18_87	Southern Suburbs	Victim	18_87	Cape Town	Victim	Rape
18_87	Southern Suburbs	Reporter	18_87	Southern Suburbs	Witness	18_87	Southern Suburbs	Witness	Forgery or Fraud
58_62	Cape Town	Reporter	58_62	Cape Town	Victim	18_87	Edgemead	Victim	Drunken Driving
18_87	Southern Suburbs	Reporter	38_42	Southern Suburbs	Reporter	18_87	Southern Suburbs	Witness	Illegal gambling
18_87	Southern Suburbs	Victim	18_87	Southern Suburbs	Reporter	18_87	Cape Town	Victim	Illegal gambling
38_42	Cape Town	Reporter	18_87	Southern Suburbs	Victim	18_87	Cape Town	Victim	Drug related
_22	Cape Town	Victim	18_22	Cape Town	Victim	18_87	Cape Town	Victim	Drunken Driving
23_27	Cape Town	Victim	18_87	Cape Town	Victim	18_87	Northern Suburbs	Victim	Arson
58_62	Cape Town	Witness	18_87	Northern Suburbs	Reporter	58_62	Cape Town	Witness	Vandalism
23_27	Cape Town	Victim	23_27	Cape Town	Victim	18_87	Southern Suburbs	Victim	Corruption
18_22	Cape Town	Proxy	18_22	Cape Town	Proxy	18_87	City Bowl	Proxy	Drunken Driving
_66	Southern Suburbs	Reporter	63_67	Southern Suburbs	Reporter	63_67	Cape Town	Reporter	Illegal gambling

Table C.2: CG-diverse sample anonymised data

A10:S5:R1			A1:S1:R1			A1:S5:R10			CRIME
48_52	Southern Suburbs	Reporter	18_87	Southern Suburbs	Witness	18_87	Cape Town	Reporter	Assault
18–87	Cape Town	Reporter	18–87	Atlantic Seaboard	Victim	18–87	Atlantic Seaboard	Victim	Forgery or Fraud
28–32	Cape Town	Witness	28–32	Atlantic Seaboard	Reporter	18–87	Atlantic Seaboard	Witness	Assault
18–87	Cape Town	Reporter	18–87	Cape Town	Reporter	18–87	Atlantic Seaboard	Reporter	Vandalism
48–52	City Bowl	Reporter	18–87	City Bowl	Victim	18–87	City Bowl	Victim	Domestic Violence
38–42	Northern Suburbs	Reporter	18–87	Northern Suburbs	Reporter	38–42	Northern Suburbs	Reporter	Disorderly conduct
18–87	Northern Suburbs	Reporter	18–87	Cape Town	Reporter	28–32	Cape Town	Reporter	Assault
28–32	Northern Suburbs	Witness	18–87	Northern Suburbs	Witness	28–32	Northern Suburbs	Witness	Forgery or Fraud
18–22	Atlantic Seaboard	Reporter	18–87	Cape Town	Reporter	18–87	Cape Town	Reporter	Rape
18–22	Northern Suburbs	Reporter	18–22	Cape Town	Witness	18–87	Northern Suburbs	Witness	Other
23–27	Cape Town	Witness	23–27	Southern Suburbs	Witness	18–87	Southern Suburbs	Witness	Disorderly conduct
–30	Cape Town	Reporter	28–32	Cape Town	Witness	18–87	Atlantic Seaboard	Witness	Illegal gambling
23_27	Cape Town	Victim	18–87	City Bowl	Victim	18–87	City Bowl	Victim	Other
18–87	Cape Town	Reporter	23–27	Cape Town	Reporter	23–27	Cape Town	Victim	Illegal gambling
58–62	Northern Suburbs	Reporter	18–87	Cape Town	Reporter	18–87	Northern Suburbs	Proxy	Murder
43–47	City Bowl	Witness	18–87	City Bowl	Witness	18–87	Cape Town	Reporter	Murder
18–22	Southern Suburbs	Witness	18–22	Southern Suburbs	Witness	18–87	Southern Suburbs	Witness	Robbery
28–32	City Bowl	Witness	18–87	Oranjezicht	Reporter	28–32	City Bowl	Witness	Domestic Violence
43–47	Northern Suburbs	Witness	18–87	Cape Town	Reporter	18–87	Goodwood	Witness	Burglary
18–87	Cape Town	Reporter	23–27	City Bowl	Reporter	18–87	Cape Town	Reporter	Arson
73–77	Cape Town	Reporter	18–87	City Bowl	Witness	18–87	City Bowl	Witness	Drunken Driving
23–27	Southern Suburbs	Reporter	23–27	Southern Suburbs	Victim	18–87	Southern Suburbs	Victim	Murder
18–87	Northern Suburbs	Victim	18–87	Cape Town	Reporter	18–87	Cape Town	Reporter	Vandalism
18–87	Atlantic Seaboard	Witness	18–87	Atlantic Seaboard	Witness	18–87	Atlantic Seaboard	Witness	Domestic Violence
18–22	Atlantic Seaboard	Reporter	18–22	Cape Town	Witness	18–87	Atlantic Seaboard	Witness	Domestic Violence
23–27	Southern Suburbs	Reporter	23–27	Southern Suburbs	Victim	18–87	Cape Town	Reporter	Forgery or Fraud
18–87	Cape Town	Reporter	18–87	Cape Town	Reporter	18–87	Northern Suburbs	Victim	Forgery or Fraud
23–27	Cape Town	Victim	18–87	Cape Town	Reporter	18–87	Atlantic Seaboard	Victim	Forgery or Fraud
43–47	Northern Suburbs	Reporter	18–87	Cape Town	Reporter	18–87	Northern Suburbs	Proxy	Domestic Violence
23–27	Southern Suburbs	Reporter	23–27	Cape Town	Reporter	23–27	Southern Suburbs	Proxy	Assault
33–37	City Bowl	Witness	33–37	Cape Town	Witness	33–37	Cape Town	Witness	Assault
18–87	Cape Town	Reporter	18–87	Cape Town	Reporter	18–87	Cape Town	Proxy	Vandalism
18–87	Northern Suburbs	Reporter	18–87	Northern Suburbs	Victim	18–87	Northern Suburbs	Victim	Domestic Violence
33–37	Cape Town	Reporter	33–37	Cape Town	Witness	33–37	Atlantic Seaboard	Witness	Drunken Driving
58–62	Southern Suburbs	Reporter	18–87	Southern Suburbs	Victim	18–87	Southern Suburbs	Victim	Rape
38–42	Southern Suburbs	Reporter	18–87	Cape Town	Reporter	18–87	Southern Suburbs	Witness	Forgery or Fraud
58–62	Northern Suburbs	Reporter	58–62	Northern Suburbs	Victim	58–62	Northern Suburbs	Victim	Drunken Driving
38–42	Southern Suburbs	Reporter	18–87	Cape Town	Reporter	18–87	Southern Suburbs	Witness	Illegal gambling
18–87	Cape Town	Reporter	18–87	Southern Suburbs	Victim	18–87	Claremont	Victim	Illegal gambling
38–42	Southern Suburbs	Reporter	18–87	Southern Suburbs	Victim	18–87	Cape Town	Reporter	Drug related
18–87	Cape Town	Reporter	18–87	Northern Suburbs	Victim	18–87	Cape Town	Victim	Drunken Driving
23–27	Cape Town	Victim	18–87	Northern Suburbs	Victim	18–87	Northern Suburbs	Victim	Arson
58–62	Northern Suburbs	Reporter	58–62	Cape Town	Reporter	18–87	Northern Suburbs	Witness	Vandalism
23–27	Southern Suburbs	Reporter	23–27	Southern Suburbs	Reporter	23–27	Cape Town	Victim	Corruption
18–22	City Bowl	Reporter	18–22	Cape Town	Proxy	18–87	Cape Town	Reporter	Drunken Driving
18_87	Cape Town	Reporter	63_67	Cape Town	Witness	18_87	Cape Town	Witness	Illegal gambling

Bibliography

[1] Aggarwal, C. C. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases* (2005), VLDB Endowment, pp. 901–909.

[2] Aggarwal, C. C. On unifying privacy and uncertain data models. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on* (2008), IEEE, pp. 386–395.

[3] Aytug, H., and Koehler, G. J. New stopping criterion for genetic algorithms. *European Journal of Operational Research 126*, 3 (2000), 662–674.

[4] Backstrom, L., Dwork, C., and Kleinberg, J. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. *Communications of the ACM 54*, 12 (2011), 133–141.

[5] Bayardo, R. J., and Agrawal, R. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on* (2005), IEEE, pp. 217–228.

[6] Bhandari, D., Murthy, C., and Pal, S. K. Variance as a stopping criterion for genetic algorithms with elitist model. *Fundamenta Informaticae 120*, 2 (2012), 145–164.

[7] Burke, M.-J., and Kayem, A. V. K-anonymity for privacy preserving crime data publishing in resource constrained environments. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on* (2014), IEEE, pp. 833–840.

[8] Byun, J.-W., Kamra, A., Bertino, E., and Li, N. Efficient k-anonymization using clustering techniques. In *Advances in Databases: Concepts, Systems and Applications*. Springer, 2007, pp. 188–200.

[9] Ciriani, V., di Vimercati, S. D. C., Foresti, S., and K-Anonymity, P. S. O. k-anonymity. *Advances in Information Security* (2007).

[10] Ciriani, V., di Vimercati, S. D. C., Foresti, S., and Samarati, P. k-anonymous data mining: A survey. In *Privacy-preserving data mining*. Springer, 2008, pp. 105–136.

[11] Coello, C. A. C. Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. *Frontiers of Computer Science in China 3*, 1 (2009), 18–30.

[12] Crawford, R., Bishop, M., Bhumiratana, B., Clark, L., and Levitt, K. Sanitization models and their limitations. In *Proceedings of the 2006 workshop on New security paradigms* (2006), ACM, pp. 41–56.

[13] Domingo-Ferrer, J., and Soria-Comas, J. From< i> t</i>-closeness to differential privacy and vice versa in data anonymization. *Knowledge-Based Systems* (2014).

[14] Dwork, C. Differential privacy. In *Automata, languages and programming*. Springer, 2006, pp. 1–12.

[15] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*. Springer, 2006, pp. 265–284.

[16] Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the 42nd ACM symposium on Theory of computing* (2010), ACM, pp. 715–724.

[17] Friedman, A., Wolff, R., and Schuster, A. Providing k-anonymity in data mining. *The VLDB Journal 17*, 4 (2008), 789–804.

[18] Fung, B., Wang, K., Chen, R., and Yu, P. S. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (CSUR) 42*, 4 (2010), 14.

[19] Fung, B. C., Wang, K., and Yu, P. S. Anonymizing classification data for privacy preservation. *Knowledge and Data Engineering, IEEE Transactions on 19*, 5 (2007), 711–725.

[20] GEHRKE, J. K. D., MACHANAVAJJHALA, A., and VENKITASUBRAMANIAM, M. l-diversity: Privacy beyond k-anonymity. In *Proc. of the 22nd IEEE International Conference on Data Engineering (ICDE), Atlanta, GA* (2006).

[21] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: an update. *ACM SIGKDD explorations newsletter 11*, 1 (2009), 10–18.

[22] Inan, A., Kantarcioglu, M., and Bertino, E. Using anonymized data for classification. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on* (2009), IEEE, pp. 429–440.

[23] Ishibuchi, H., Nakashima, T., and Murata, T. Comparison of the michigan and pittsburgh approaches to the design of fuzzy classification systems. *Electronics and Communications in Japan(Part III Fundamental Electronic Science) 80*, 12 (1997), 10–19.

[24] Iyengar, V. S. Transforming data to satisfy privacy constraints. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (2002), ACM, pp. 279–288.

[25] Jang, M.-H., Kim, S.-W., Faloutsos, C., and Park, S. Accurate approximation of the earth mover's distance in linear time. *Journal of Computer Science and Technology 29*, 1 (2014), 142–154.

[26] Kabir, M. E., Wang, H., and Bertino, E. Efficient systematic clustering method for k-anonymization. *Acta Informatica 48*, 1 (2011), 51–66.

[27] Kifer, D., and Gehrke, J. Injecting utility into anonymized datasets. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (2006), ACM, pp. 217–228.

[28] Konak, A., Coit, D. W., and Smith, A. E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety 91*, 9 (2006), 992–1007.

[29] Last, M., Tassa, T., Zhmudyak, A., and Shmueli, E. Improving accuracy of classification models induced from anonymized datasets. *Information Sciences 256* (2014), 138–161.

[30] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (2005), ACM, pp. 49–60.

[31] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on* (2006), IEEE, pp. 25–25.

[32] Levina, E., and Bickel, P. The earth mover's distance is the mallows distance: some insights from statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (2001), vol. 2, IEEE, pp. 251–256.

[33] Li, N., Li, T., and Venkatasubramanian, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE* (2007), vol. 7, pp. 106–115.

[34] Lin, J.-L., and Wei, M.-C. An efficient clustering method for k-anonymization. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society* (2008), ACM, pp. 46–50.

[35] Lin, J.-L., and Wei, M.-C. Genetic algorithm-based clustering approach for< i> k</i>-anonymization. *Expert Systems with Applications 36*, 6 (2009), 9784–9792.

[36] Ling, H., and Okada, K. An efficient earth mover's distance algorithm for robust histogram comparison. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 5 (2007), 840–853.

[37] Lukasiewycz, M., Glass, M., Reimann, F., and Teich, J. Opt4J - A Modular Framework for Meta-heuristic Optimization. In *Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 2011)* (Dublin, Ireland, 2011), pp. 1723–1730.

[38] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD) 1*, 1 (2007), 3.

[39] Mackey, E., and Elliot, M. Understanding the data environment. *XRDS: Crossroads, The ACM Magazine for Students 20*, 1 (2013), 36–39.

[40] Matatov, N., Rokach, L., and Maimon, O. Privacy-preserving data mining: A feature set partitioning approach. *Information Sciences 180*, 14 (2010), 2696–2720.

[41] Morton, S., Mahoui, M., and Gibson, P. J. An automated data utility clustering methodology using data constraint rules. In *Proceedings of the 2012 international workshop on Smart health and wellbeing* (2012), ACM, pp. 9–16.

[42] Nergiz, M. E., and Clifton, C. Thoughts on k-anonymization. *Data & Knowledge Engineering 63*, 3 (2007), 622–645.

[43] Nergiz, M. E., Tamersoy, A., and Saygin, Y. Instant anonymization. *ACM Transactions on Database Systems (TODS) 36*, 1 (2011), 2.

[44] Rokach, L. Genetic algorithm-based feature set partitioning for classification problems. *Pattern Recognition 41*, 5 (2008), 1676–1700.

[45] Rubner, Y., Tomasi, C., and Guibas, L. J. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision 40*, 2 (2000), 99–121.

[46] Shmueli, E., and Tassa, T. Privacy by diversity in sequential releases of databases. *Information Sciences* (2014).

[47] Shmueli, E., Tassa, T., Wasserstein, R., Shapira, B., and Rokach, L. Limiting disclosure of sensitive data in sequential releases of databases. *Information Sciences 191* (2012), 98–127.

[48] Spiekermann, S. The challenges of privacy by design. *Communications of the ACM 55*, 7 (2012), 38–40.

[49] Sweeney, L. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10*, 05 (2002), 557–570.

[50] Terblanche, S. Sentencing guidelines for south africa: Lessons from elsewhere. *South African Law Journal 120*, 4 (2003), p–858.

[51] Terrovitis, M., Mamoulis, N., and Kalnis, P. Privacy-preserving anonymization of set-valued data. *Proceedings of the VLDB Endowment 1*, 1 (2008), 115–125.

[52] Venkatasubramanian, S. Moving heaven and earth: Distances between distributions. *ACM SIGACT News 44*, 3 (2013), 56–68.

[53] Wang, K., and Fung, B. Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), ACM, pp. 414–423.

[54] Whitley, L. D., et al. The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *ICGA* (1989), vol. 89, pp. 116–123.

[55] Wicker, S. B. The loss of location privacy in the cellular age. *Communications of the ACM 55*, 8 (2012), 60–68.

[56] Xiao, X., and Tao, Y. Anatomy: Simple and effective privacy preservation. In *Proceedings of the 32nd international conference on Very large data bases* (2006), VLDB Endowment, pp. 139–150.

[57] Xu, J., Zhang, Z., Tung, A. K., and Yu, G. Efficient and effective similarity search over probabilistic data based on earth mover's distance. *Proceedings of the VLDB Endowment 3*, 1-2 (2010), 758–769.

[58] Xu, Y., Wang, K., Fu, A. W.-C., and Yu, P. S. Anonymizing transaction databases for publication. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), ACM, pp. 767–775.

[59] Zhang, Q., Koudas, N., Srivastava, D., and Yu, T. Aggregate query answering on anonymized tables. In *ICDE* (2007), vol. 7, Citeseer, pp. 116–125.

[60] Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L., and Thiele, L. Spea2: Improving the strength pareto evolutionary algorithm, 2001.